Compilador Assembler no TRS-80 com QUASARIV, um jogo-exemplo.

ANO IV – Nº42 – MARÇO – Cr$ 5.000

Micro Sistemas

A PRIMEIRA REVISTA BRASILEIRA DE MICROCOMPUTADORES

A COMUNICAÇÃO DOS DADOS

A SYSDATA GANHA DINHEIRO FAZENDO MICROCOMPUTADORES COMO O SYSDATA III.

ALGUMAS PESSOAS GANHAM DINHEIRO COMPRANDO.

SYSDATA III
Aqui, tudo o que Você espera de um grande micro.

Compatível com o TRS-80 Modelo III da Radio Shack.
Gabinete, teclado e CPU em módulos independentes.
Versões de 64 a 128 KBytes de RAM, 16 KBytes de ROM.
Teclado profissional com numérico reduzido e 4 teclas de funções.
Sistema operacional de disco DOS III ou CP/M 2.2.
Caracteres gráficos.
Vídeo composto com 18 MHZ de faixa de passagem.
Saída para impressora paralela.

SYSDATA III
Software disponível variado. Escolha o seu.

Videotexto (TELESP).
Projeto Cirandão (EMBRATEL).
Rede de telex.
Sistema Gerenciador de Banco de Dados (SGBD), DBASE II.
Compiladores Cobol, Fortran, Pascal, Basic, Forth, Lisp e Pilot.
Editor de textos. Editor de Assembler.
Desassemblador.
Debugador.
Visicalc.
Wordstar,
e muitos outros.

SYSDATA III
Características técnicas. Para aqueles que querem saber tudo.

Total compatibilidade com o TRS-80 Modelo III da Radio Shack.
Processador Z-80-A.
Vídeo de 16 x 64 ou 16 x 32 (linhas x colunas).
Alimentação de 110 V ou 220 V.
Teclado alfanumérico de 69 teclas.
Teclado numérico reduzido com 4 teclas de funções.
Gráficos com 128 x 48 pontos no vídeo.
Aceita até duas RS-232-C (Síncronas ou Assíncronas).
Modem (opcional).
Saída paralela para impressora.
Placa controladora para até 4 drives de 5 e 1/4", dupla densidade (180 KBytes por face), face simples (dupla face opcional).

Opções futuras:

Vídeo compatível 16 x 64, 16 x 32, 24 x 80 ou 24 x 40 (linhas x colunas).
Expansão até 256 KBytes de RAM.
Alta resolução gráfica e cor.
Interface para acionamento de disco rígido (Winchester) de 5, 10 ou 20 MBytes.
Clock dobrado (4,0 MHZ).
Total compatibilidade com o TRS-80 Mod. IV.
CP/M versão 3.0.

SYSDATA

Sysdata eletrônica ltda. 01155 - Av. Pacaembú, 788 - Pacaembú - SP - Fone: (011)826.4077

Ano IV
Nº 42
Março 1985

Micro Sistemas

SUMÁRIO

P&D Sistemas Eletrônicos S.A.
Biblioteca

10 A VIAGEM DOS DADOS
Neste artigo, Roberto Quito de Sant'Ana comenta que o grande assunto do momento nas rodas de hobbystas e usuários de micros é a transmissão de dados, ou simplesmente a comunicação entre os equipamentos.
Em linguagem acessível, ele explica os sistemas em funcionamento no Brasil e dá uma visão panorâmica a respeito de como se processam a saída e a entrada dos dados nas máquinas.

30 QUASAR IV, UMA AVENTURA COMPILADA
O usuário vai conhecer neste artigo toda a profundidade do Quasar IV, um jogo cuja principal característica é fugir da fórmula do interpretador, considerado monótono ou muito lento por alguns. Trata-se, segundo Lávio Pareschi, de um passatempo com múltiplas opções, que ora exigem sorte, às vezes malandragem, quando não muita habilidade. Um jogo fácil, difícil, desafiante.

52 ARQUIVOS EM DISCO DO NEWDOS/80
Conclusão do artigo cuja primeira parte foi publicada em MS 39.
Nesta última parte João Henrique Volpini Mattos ensina a praticar os novos comandos utilizados com os arquivos NEWDOS/80, de maneira simples, fazendo com que o usuário perca o natural temor de se aventurar por caminhos que alguns consideram complicados, como esses arquivos.

62 OS PERIGOS DA TELEMÁTICA
A máquina pensa ou não? Bem, este assunto e outras questões de profunda **subjetividade** filosófica são expostos e comentados de forma bem humorada por Luís Carlos Eiras, em mais um conto em que a informática é o ponto central. Ele narra as experiências de um usuário que se aventura a utilizar seu equipamento em busca de contatos com outros seres terrenos durante a madrugada.

20 AUTOMAÇÃO: UM CAMINHO PARA AS REDES LOCAIS - Como são e quais as vantagens das redes locais. Veja neste artigo de Amaury Moraes Junior.

26 MODEMS, UM PERIFÉRICO EM VOGA - Uma abordagem abrangente acerca desse importante periférico na comunicação de dados. Reportagem.

48 APPLE, O MAPA DA ROM - Aldo Felício Naletto Junior, na primeira parte de seu artigo, começa a explicar o mapa da ROM do Apple.

BANCO DE SOFTWARE
* 64 - Polvo Gigante * 66 - Curvas Fantásticas
* 69 - Lista Telefônica * 72 Solitário

SEÇÕES

4 EDITORIAL	24 BITS	76 CLASSIFICADOS
6 CARTAS	74 DICAS	78 LIVROS

editorial

A esta altura do campeonato, você já deve estar sentindo leves ventos de mudanças no perfil de **MS**. *É bem verdade que ainda é cedo para julgar se tais mudanças são boas ou ruins, porém tenho certeza de que, em dois ou três meses, estaremos às voltas com um batalhão de cartas, dos mais variados pontos do Brasil, cada qual trazendo, no mínimo, uma sugestão ou crítica.*

Seria um exagero dizer que são essas cartas que nos levam ao caminho A ou B, mas certamente elas constituem parte fundamental de nosso combustível. A partir delas, a gente reflete bastante, discute — e como — e decide manter ou alterar o rumo. Às vezes não conseguimos nos esquecer, mesmo em casa, nos fins-de-semana, dos elogios apaixonados ou das críticas ferozes.

É imensa a responsabilidade de ter um grupo de leitores tão atentos. Não nos permite a inércia, jamais. Também o fato de estarmos há quatro anos batalhando neste mercado nos deu experiência suficiente para fugir da acomodação. Durante este tempo, MICRO SISTEMAS esteve sempre inovando; levando ao leitor importantes informações e, principalmente, servindo de ponte entre o usuário e a indústria. Essa foi nossa maior preocupação: criar condições para que nossos leitores vivessem plenamente os recursos oferecidos pelo mercado brasileiro de microinformática.

Mas nós vamos mudar. O que era bom em **MS**, *trabalharemos para que fique ainda melhor, pois faremos de 85 o ano do usuário, do leitor. Iremos reestruturar algumas seções e serviços e procuraremos agilizar nosso esquema de atendimento às dúvidas.*

Para os que se desanimam perante dez páginas de uma (boa) listagem, aconselhamos um pouco de paciência, pois estamos preparando o **MS Save**, *para diminuir os entraves da digitação. Os que reclamaram a ausência de nosso tradicional Índice MS podem estar tranqüilos, pois vem aí o* **Acesso Direto**, *um resumo completo destes três anos de MICRO SISTEMAS por edição, assunto e linha de equipamento. Outro serviço, as* **Micro Fichas**, *será bastante útil para acabar de vez com os papiros-lembretes. Quanto aos cursos periodicamente apresentados por* **MS**, *estamos estudando os pedidos e a viabilidade de produção. Continuem a enviar sugestões.*

É isto. Neste mês histórico, em que se inicia uma nova era para nosso país, anunciamos, também para **MS**, *um novo ciclo, cujo sucesso, de maneira idêntica, será função direta da participação de todos.*

Alda Campos

CAPA:
Roberto De Vicq

Micro Sistemas

EDITOR/DIRETOR RESPONSÁVEL:
Alda Cristina Surerus Campos

DIRETOR-TÉCNICO:
Renato Degiovani

ASSESSORIA TÉCNICA: Roberto Quito de Sant'Anna; José Eduardo Neves; Orson V. Galvão; Luiz Antonio Pereira; Heloisa Ferreira

CPD: Pedro Paulo Pinto Santos (responsável)

REDAÇÃO: Graça Santos (Subeditoria); Beni Lima Pereira; Cláudia Salles Ramalho; Maria da Glória Esperança; Stela Lachtermacher

COLABORADORES: Amaury Moraes Jr.; Antonio Costa Pereira; Armando Oscar Cavanha Filho; Carlos Alberto Diz; Esdras Avelino Leitão; Evandro Mascarenhas de Oliveira; Heber Jorge da Silva; Ivo D'Aquino Neto; João Antonio Zuffo; João Henrique Volpini Mattos; Jorge de Rezende Dantas; José Carlos Niza; José Ribeiro Pena Neto; José Roberto França Cottim; Lávio Pareschi; Luciano Nilo de Andrade; Luis Lobato Lobo; Luis Carlos Eiras; Luiz Gonzaga de Alvarenga; Marcel Gameleira de Albuquerque; Maurício Costa Reis; Paulo Sérgio Gonçalves; Rizieri Maglio; Rudolf Horner Jr.; Sérgio Veludo

ARTE: Marta Heilborn (coordenação); Leonardo A. Santos (diagramação); Maria Christina Coelho Marques (revisão); Wellington Silvares (arte final).

ACOMPANHAMENTO GRÁFICO: Fábio da Silva

ADMINISTRAÇÃO: Janete Sarno

PUBLICIDADE
São Paulo:
Natal Calina
Contatos: Eloisa Brunelli; Marisa Coan; Paulo Gomide.
Tels.: (011) 853-3229
853-3152

Rio de Janeiro:
Elizabeth Lopes dos Santos
Contatos: Regina de Fátima Gimenez; Georgina Pacheco de Oliveira

Minas Gerais:
Representante: Sidney Domingos da Silva
Rua dos Caetés, 530 — sala 422
Tel.: (031) 201-1284, Belo Horizonte

CIRCULAÇÃO E ASSINATURAS:
Ademar Belon Zochio (RJ)

COMPOSIÇÃO:
Gazeta Mercantil S/A Gráfica e Comunicações
Studio Alfa
Coopim

FOTOLITO:
Organização Beni Ltda.
CHD Composição Ltda.
Studio gráfico GL.

IMPRESSÃO:
JB Indústrias Gráficas

DISTRIBUIÇÃO:
Fernando Chinaglia Distribuidora Ltda.
Tel.: (021) 268-9112

ASSINATURAS:
No país: 1 ano — Cr$ 50.000

Os artigos assinados são de responsabilidade única e exclusiva dos autores. Todos os direitos de reprodução do conteúdo da revista estão reservados e qualquer reprodução, com finalidade comercial ou não, só poderá ser feita mediante autorização prévia. Transcrições parciais de trechos para comentários ou referências podem ser feitas, desde que sejam mencionados os dados bibliográficos de **MICRO SISTEMAS**. A revista não aceita material publicitário que possa ser confundido com matéria redacional.

MICRO SISTEMAS é uma publicação mensal da

Análise, Teleprocessamento e Informática Editora Ltda.

Endereços:
Rua Oliveira Dias, 153 - Jardim Paulista - São Paulo/SP - CEP 01433 - Tels.: (011) 853-3800 e 881-5668.

Av. Presidente Wilson, 165 - grupo 1210 - Centro - Rio de Janeiro/RJ - CEP 20030 - Tels.: (021) 262-5259, 262-6437 e 262-6306.

A COMPUMICRO JÁ TEM O MICROENGENHO 2.

A Compumicro é a única empresa do Rio que comercializa exclusivamente micros para uso profissional, em um amplo e confortável escritório.

Com uma equipe de vendas formada somente por profissionais de informática, a Compumicro vem se destacando como uma das maiores e mais bem preparadas empresas do setor.

Isto se deve ao fato da Compumicro oferecer um atendimento altamente especializado, só comercializando equipamentos de qualidade comprovada.

Como um sucesso puxa o outro, a Compumicro coloca a sua disposição o Microengenho 2. O único micro computador nacional totalmente compatível com APPLE IIe americano.

O Microengenho 2 gera caracteres em português maiúsculos, minúsculos e acentuados a partir do teclado. Pode ter resolução gráfica de 107.520 pontos no vídeo (dobro do APPLE II Plus). E ainda possibilita o uso de uma placa de modem 1275A, operando em modo FULL-DUPLEX (cirandão) e HALF-DUPLEX (MicroxMicro) substituindo o modem externo convencional.

Venha a Compumicro e comprove este sucesso pessoalmente.

Compumicro. O melhor em micro pelas melhores condições.

GARANTIA DE 1 ANO
GRÁTIS EDITEX III MICROCÁLCULO II

SUCESSO PUXA SUCESSO.

SPECTRUM
MICRO engenho²

compumicro
INFORMÁTICA EMPRESARIAL LTDA.
End.: Rua Sete de Setembro, 99 - 11.° andar - Tel.: PABX (021) 224-7007
CEP 20050 - Rio de Janeiro/RJ.

cartas

O sorteado deste mês, que receberá uma assinatura de um ano de MICRO SISTEMAS, é Antonio Roberto Barrichello, de São Paulo.

RESPOSTA AO *GARIMPANDO...*

Sou possuidor de um TK-85 com 16 K de memória, assim como o leitor Ricardo Mendonça, que relatou sua experiência na Seção Cartas de MS nº 34, na carta intitulada *"Garimpando* bytes". Quero, se possível, esclarecer a dúvida do Ricardo: no que tange ao funcionamento interno do microcomputador, pouco sei, mas posso assegurar, todavia, que o processador Z80 é um processador de 8 bits, que permite o agrupamento de dois registradores internos de modo a formar uma palavra de 16 bits.

Os 16 bits recebem as seguintes denominações: A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0. Assim, o processador tem condições de acessar 65536 (2^{16} ou 64 K) bytes de memória (ou posições). Porém, o Sistema Operacional dos micros da linha Sinclair seta o bit A15 (veja em MS nº 31, pág. 42). Desta forma, o processador só pode acessar 32768 (2^{15} ou 32 K) posições de memória.

Por causa desta particularidade do Sistema Operacional da linha Sinclar, os números maiores que 32767 ($2^{15}-1$) são *vistos* pelo Sistema Operacional da seguinte forma: N-32768, porque quem define se um número é maior ou menor que 32767 é o bit A15. Se ele não for considerado, haverá uma mera repetição dos números de 0 a 32767. E se dermos um **POKE** no endereço 57344, como o leitor Ricardo Mendonça fez, estaremos, na realidade, dando um **POKE** no endereço 24577.

Para chegar a esta conclusão usei os comandos **B, E** e **M** do MICRO BUG, e pesquisei os endereços de 32757 até 32767 e os endereços de 65525 até 65535.
O resultado foi:
32757 48 65525 48
32758 A6 65526 A6
32759 0D 65527 0D
e assim sucessivamente. Podemos reparar que o endereço da esquerda é igual ao da direita, diminuindo-se 32768. Se pegarmos, por exemplo, os dois últimos números que pesquisei (32767 3E e 65535 3E) e transformá-los em binário, teremos:
32767= 0111 1111 1111 1111
65535= 1111 1111 1111 1111

A única diferença entre os dois números binários é o bit mais significativo (A15).

Ainda usando o MICRO BUG, criei a linha 1 **REM** com 99 caracteres e entrei com a dica "Surpresa na tela" (Seção Dicas de MS nº 34), só que não usei o endereço 16514, mas sim o endereço 49282 (16514+32768). Depois retornei ao BASIC, testei a rotina com **RAND USR** 16514 e ela funcionou perfeitamente. Tentei acessar a rotina através de **RAND USR** 49282 mas não funcionou, dando notação 0/0. O que deve ter acontecido com o leitor Ricardo é que provavelmente ele utilizou um programa BASIC para fazer a constatação citada na carta e, por sorte ou azar (não sei), isto não interferiu no programa.

Gilberto F. da Silva
São Bernardo do Campo-SP

Agradecemos a você, Gilberto, e também a diversos leitores que nos escreveram explicando o que realmente ocorreu com o micro do Ricardo.

CONVERSA DE PROGRAMADORES

Recebemos em nossa redação a colaboração espontânea do nosso amigo leitor Belmiro, em que este faz alguns comentários irônicos sobre o programa "PIL, a fertilidade programada", publicado em MS nº 31, pág. 32.

Dizem que já aconteceu (sobre o programa Pil, de Armando Oscar e Maria Beatriz Cavanha):

Um programador encontra-se com seu amigo, que vinha usando o PIL desde o casamento.

— Olá, como vai? Puxa, há quanto tempo a gente não se vê, cara! Quem é esse menininho?

— É o meu ET2.

— ET2?!

— É... Erro de Tabela 2. Aconteceu num espaço de sete anos. Meu ET1 já vai fazer nove anos: ocorreu no segundo ano de aplicação. Depois eu melhorei a performance.

— Então, daqui a uns 12 anos sai o ET3.

— Prá mim chega. Já mudei a técnica. Abandonei o "software" e apliquei o "hardware".

— O COMPUTADOR... cara?!

— O bisturitex.

Belmiro F. da Silva
Rio de Janeiro — RJ

I CHING NO CP-300

Na revista MS nº 26, foi publicado o programa *I CHING*, mas quando fui rodá-lo no meu CP-300, deu erro nas linhas 130, 140, 150 e 160: toda vez que pressionava **RUN** dava erro nestas linhas. Um outro problema ocorre nas linhas 465 a 475, com a mensagem de erro: "subscrito fora de faixa" na linha 475. O valor de **G** na variável **J$**, na linha 475, fica entre 500 e 600. E o erro que está acontecendo nas linhas 130, 140, 150 e 160 é um erro de sintaxe, pois o BASIC sem Disco não aceita a instrução:

```
NL MID$(D$,L,1)="1"
```

que deve ser mudada para:

```
NL K$=MID$(D$,L,1):K$="1"
```

Gostaria que MS entrasse em contato com o autor para a solução dos problemas citados, ou seja, modificar as linhas 130, 140, 150, 160, 370, 385, 390, 400, 410 e 420, permitindo que mesmo quem não tenha disco possa desfrutar do *I CHING*.

Gerson Petrucelli Filho
São Carlos-SP

Remetemos a sua carta para o nosso amigo Luiz Gonzaga de Alvarenga, autor do programa *I CHING*, e eis a resposta que recebemos:

"Realmente o CP-300 não aceita a atribuição direta da função *string* MID$, pois esta é exclusiva do BASIC Disco.

O valor de G que você encontrou é igual a 517, e é decorrente da atribuição de variável ocorrida na linha 40, com decrementos de 64 em 64, nas linhas 130, 140, 150 e 160. A modificação apresentada em sua carta não é suficiente. O que ocorreu é que, na linha 440, não foi encontrada a *string* H$+T$ (comparação feita na linha 445) para que fosse feita uma nova atribuição de variável, onde G tomasse o valor de A (que seria, no máximo, igual a 8). Naturalmente, o valor de G na linha 475 manteve o seu último valor, o que acarretou erro de dimensionamento.

Apresento, a seguir, as modificações que se podem efetuar para que o programa rode no CP-300:

```
102 K$(L)=MID$(D$,L,1)
105 IFK$(L)="1" ...
110 IFK$(L)="2" ...
115 IFK$(L)="3" ...
120 IFK$(L)="4" ...
130 K$(L)="1" ...      ...IFL)6THEN160ELSE102
140 K$(L)="0" ...      ...IFL)6THEN160ELSE102
150 K$(L)="1" ...      ...IFL)6THEN160ELSE102
160 K$(L)="0" ...      ...IFL)6THEN160ELSE102
165 GOTO102
168 FORWQ=1TO6:AF$=AF$+K$(WQ):NEXT
169 D$=AF$
367 F$(K)=MID$(T$,K,1)
370 IFF$(K)="1" ...
375 IFF$(K)="2" ...
380 IFF$(K)="3" ...
385 IFF$(K)="4" ...
390 F$(K)="1" ...                    ...ELSE367
400 F$(K)="0" ...                    ...ELSE367
410 F$(K)="0" ...                    ...ELSE367
420 F$(K)="1" ...                    ...ELSE367
432 FORWQ=1TO6:FA$=FA$+F$(WQ):NEXT
433 T$=FA$
```

Luiz Gonzaga de Alvarenga
Goiânia-GO

CONTROLE DE CARGAS ELÉTRICAS

Na revista nº 20, de maio de 1983, foi publicado um artigo que me interessou: "TK e NE no Controle de Cargas Elétricas". Sendo possuidor de um TK 82-C (versão nova), estudei e montei o circuito, porém este não funcionou como o previsto.

Após ligar a interface no micro, ocorria o seguinte: ao digitar o programa tudo ficava estável e, logo depois, ao introduzir a variável **A**, a saída oscilava como se, de repente, rapidamente, muitos endereços tivessem sido liberados. Em seguida, porém, ficava estável mas sempre com o mesmo endereço. Depois disso, cada toque do teclado correspondia a uma mudança para **F(H)** nos bits menos significativos do endereço.

Para facilitar a visualização das saídas, liguei um CI-9368 ao CI-8212 e um display FND-560. Usei também, para segurança no funcionamento da interface, uma fonte de alimentação usando o CI-7805 e, logicamente, interliguei o terra como o do micro, mas a situação não mudou.

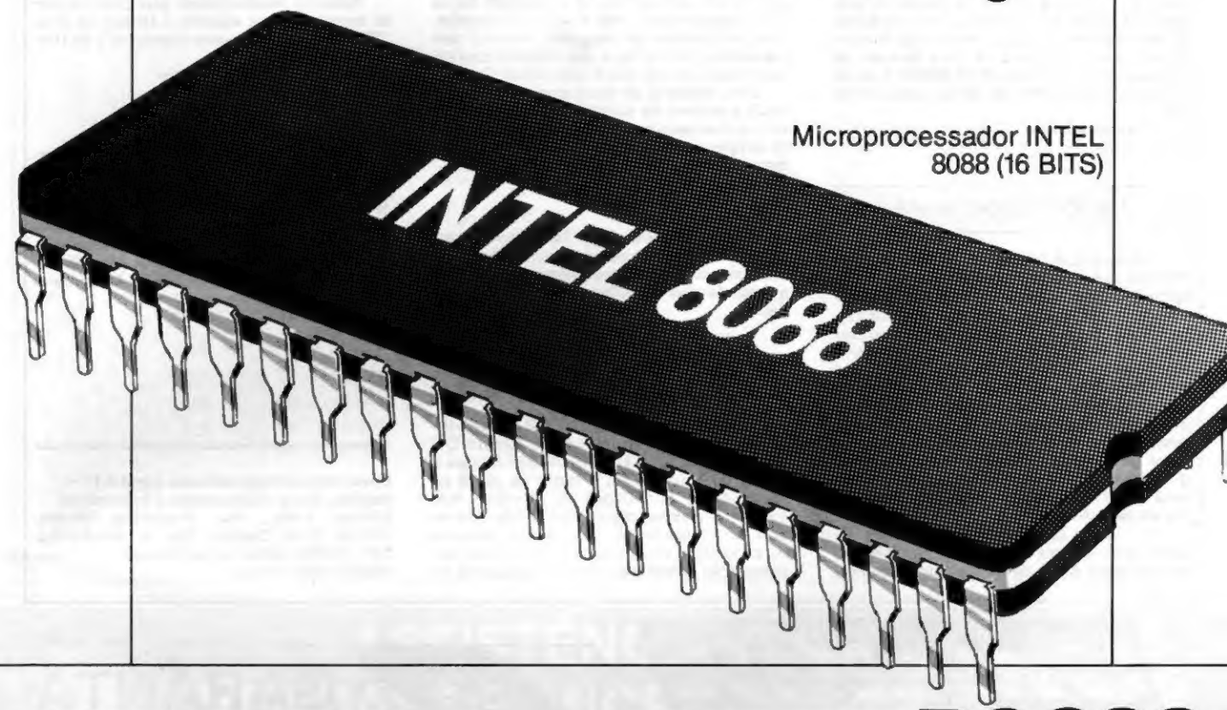
Ficaria muito grato se o autor fosse consultado para dar o seu parecer com relação ao ocorrido: aconteceu alguma errata na matéria ou o circuito só funciona em outro micro?

Milton Vilela
São Paulo — SP

Remetemos a sua carta, Eng. Milton, para o nosso amigo e autor do artigo, Jerre Palmeira Salles. Ele respondeu o seguinte:

"Esta questão é *sui gêneris*. Até agora não havia aparecido nenhuma carta com este problema. Na minha opinião, só existe uma possibilidade para o ocorrido: o barramento de dados é compartilhado com outras atividades do microprocessador, ou seja, durante um certo tempo ele recebe sinal, em seguida transmite sinal e depois refresca a memória. Se durante os períodos em que o microprocessador está enviando sinais outra fonte também estiver usando este barramento, haverá um conflito de informações e aparecerá o que você viu na tela.

COMPUMICRO
Nós dominamos esta tecnologia.



Microprocessador INTEL
8088 (16 BITS)

nexus1600 PC2001

Só quem domina esta tecnologia pode oferecer o que há de melhor em 16 Bits

- CPU'S Standard 256 K
- Drives 5 ¼ DFDD (360 K)
- Winchester de 5 e 10 MB
- Monitores cromáticos/mono
- Co-processador 8087
- Expansões de memória
- Todos os modelos de impressora
- Emulação de terminais / RJE
- Comunicação micro x mainframe
- Sistemas multiusuário
- Conversores de protocolo
- Redes locais
- Software nacional e estrangeiro

Além disso, a Compumicro oferece com exclusividade o dispositivo **8088 processor card** que permite operar software da linha **PC** em micros da linha **Apple.**

Venda, leasing e aluguel em 12, 18 e 24 meses com opção de compra.
O maior revendedor Nexus 1600 e PC 2001 do país.
PRONTA ENTREGA

compumicro
INFORMÁTICA EMPRESARIAL LTDA.
Rua Sete de Setembro, 99 - 11.º andar
Tels: PBX (021) 224-7307 - 224-7007 - RJ

Esta duplicidade de informação poderia ser:
1 — Curto-circuito entre os pinos do conector que você usou e a saída de expansão do TK.
2 — Defeito no 8212 e ele está curto-circuitando o barramento de dados ou o de endereços.

Sugiro que você confira as conexões do item 1 e verifique se não há ligação errada, tanto na fiação da expansão quanto do 8212. A interface abordada em meu artigo funciona em qualquer micro da linha Sinclair (o meu equipamento é um NEZ-8000). E as informações divulgadas na revista estão corretas."
Jerre Palmeira Salles
Crato — CE

NAMORANDO COM MS

Confesso que foi através de MICRO SISTEMAS que me apaixonei pelos micros. Desejo também confessar a minha personalidade *volúvel*, já que antes de por os olhos nesta revista, eu era um grande admirador dos grandes sistemas, pois sou um *aspirante a programador* COBOL. Mas pouco a pouco me deixei levar pela *graça*, rapidez e simplicidade do BASIC e dos micros. E esta revista me possibilitou conhecer e me aproximar desta minha nova *paixão*. Creio que como leitor de outras publicações do gênero posso dizer que MICRO SISTEMAS é a melhor revista sobre Informática deste país: vocês estão de parabéns.

Mas, como cedo ou tarde um pouco daquela arrebatadora emoção do *primeiro encontro* passa e nos deixa raciocinar melhor, me vejo agora no direito (que aliás, não sei de onde tirei) de fazer algumas reivindicações para dar um pouco mais de colorido a este meu *namoro*: se for possível, publiquem mais cursos de programação (FORTH, MUMPS, Pascal...), pois é a melhor forma de podermos seguir a rápida evolução da comunicação programadores-sistemas.

Peço também mais programas voltados para a área de cálculos e problemas sérios (com respectivos fluxogramas) e, quem sabe, uma seçãozinha de hardware, mesmo que pequenina. Isto porque sou também apaixonado (que volúvel, não?) pela eletrônica.

Bem, desde já os meus agradecimentos a vocês e podem ter certeza que o meu *namoro*, e o dos meus companheiros leitores, estará sempre aceso enquanto pudermos ver nas bancas a nossa MICRO SISTEMAS.
Marcos A. Pires
Mogi das Cruzes — SP

Ótimo, Marcos. Aqui todo mundo *gamou* por sua carta: gente como você nos dá uma alegria especial por nosso trabalho. E quanto às suas sugestões, estão todas anotadas.

NEWDOS

Ótimo o artigo "O NEWDOS que não está nos manuais", subscrito por Renato Degiovani, publicado em MS nº 31. Apenas a título de informação, o autor na parte de Manipulação dos Dados do Diretório não menciona a reparação do GAT. Não obedecida esta providência, na próxima gravação de programas no disquete poderá haver superposição desastrosa sobre o programa recuperado.

Embora o reparo no GAT possa ser feito usando os recursos do SUPERZAP, o mais prático e seguro será a gravação do programa recuperado em outro disquete. O programa poderá também, eventualmente, ser regravado no disquete-teste, desde que se tome cuidado de digitar o nome do programa e a extensão de forma idêntica às originais.

Bastante recomendável para quem quiser se aprofundar no assunto a leitura do livro "TRS-80 Disk and other mysteries", de Harvard C. Pennington.
Antonio Roberto Barrichello
Piracicaba — SP

Agradecemos a atenção, você está correto, realmente houve esta falha no texto, se bem que o DIRCHECK continuaria a apresentar o problema. Com relação à sugestão para consertar esta falha, o autor, Renato Degiovani, discorda da solução apresentada, pois, segundo ele, com a monitoração do DIRCHECK o uso do SUPERZAP torna-se bastante seguro.

Envie suas correspondências para: ATI — Análise, Teleprocessamento e Informática Editora Ltda., Av. Presidente Wilson, 165/gr. 1210, Centro, Rio de Janeiro/RJ, CEP 20030, Seção Cartas/Redação MICRO SISTEMAS.

CIBERNE SOFTWARE

apresenta novas fitas com desafios emocionantes para você!

PARA EQUIPAMENTOS COM LÓGICA SINCLAIR

1. VALKIRIE
Pilote a nave Valkirie e parta em busca de dez castelos perdidos. (Exclusividade Ciberne, por Divino C.R. Leitão).
É mais: GUERRILHA COSMICA e ZOR.

2. MERCADOR DOS SETE MARES
No século XIX você percorre o mundo a bordo de seu navio, em busca de bons negócios.
É mais: CORRIDA MALUCA e PINBALL (Exclusividade Ciberne, por Divino C.R. Leitão).

3. SUBESPAÇO
Implacável caçada espacial. Totalmente gráfico.
É mais: CAVERNAS DE MARTE (Exclusividade Ciberne, por Divino C.R. Leitão) e COMBOIO ESPACIAL.

4. DEFENSOR 3D
Livre nosso planeta de uma invasão alienígena. Fantásticas simulações tri-dimensionais.
É mais: CYBERT (Exclusividade Ciberne, por Divino C.R. Leitão) e ASSALTO.

5. ROT I - PLUS
• S.O.G.
Sistema operacional, com linguagem gráfica. Infinitas opções de uso. Totalmente em código de máquina (Exclusividade Ciberne, por J. Magel).
• MERGE
Possibilita a junção de vários programas, uns aos outros.

6. APLIC'1
• COMP-CALC
Rápido, eficiente e totalmente em código de máquina. A melhor versão do já famoso Visi-Calc.
• COMP-ARQ
Programa gerador de arquivos. Totalmente em código de máquina. Modele fichas e as acesse pelo campo que quiser.
• COMP-TEXTO
De fácil manipulação, totalmente em código de máquina.

PARA EQUIPAMENTOS COM LÓGICA TRS-80

1. SIMULADOR DE VÔO
Totalmente gráfico e acompanhado de livro de instruções, com diagramas, tabelas etc.
É mais: PINTOR MALUCO e O DESAFIO DA GALINHA.

2. XADREZ
O mais tradicional dos jogos, reeditado em nova e brilhante versão.
É mais: PATRULHA ARMADA e PÂNICO (totalmente sonorizados).

ADQUIRA ESSES LANÇAMENTOS NO SEU REVENDEDOR CIBERNE MAIS PRÓXIMO. E TAMBÉM: Bichos e Cia., Patrulha Galáctica, Aventura e Mistério, Combate, Rot II e Comando 3D (fita virgem).

JVA MICROCOMPUTADORES LTDA.
Distribuição e Informações:
Tel.: (021) 262-6569
Av. Graça Aranha, 145 - sobreloja 01
Rio de Janeiro - RJ - CEP 20030

Micro Sistemas

GARANTA SUA MS TODO MÊS!

ASSINE HOJE MESMO E RECEBA GRATUITAMENTE 6 NÚMEROS À SUA ESCOLHA A PARTIR DO Nº 13. PREENCHA O CUPOM ABAIXO (OU UMA XEROX, CASO VOCÊ NÃO QUEIRA CORTAR A REVISTA):

Nome _____
Empresa _____
Profissão/Cargo _____
Endereço para remessa _____
Cidade _____ CEP _____ Estado _____
Assinatura Anual: ☐ Micro Sistemas . . . Cr$ 50.000,00

GRÁTIS! 6 NÚMEROS ATRASADOS.

Preencha um cheque nominal À ATI Editora Ltda., e envie para: Av. Presidente Wilson, 165/Grupo 1210, Centro, Rio de Janeiro, RJ, CEP 20.030 — Tels.: (021) 262-5259 e 262-6306. R. Oliveira Dias, 153, Jardim Paulista, São Paulo, SP, CEP 01433 — Tels.: (011) 853-3574 e 853-3800.
Seu recibo será enviado pelo Correio.

A Compumicro vai deixar você com a melhor impressão do Unitron AP II

Não existe nada mais pessoal do que uma impressão digital. Ela é única. Ninguém tem igual. O mesmo acontece quando você compra o seu UNITRON AP II na COMPUMICRO.

Aqui você tem um atendimento personalizado e exclusivo.

O que este atendimento tem de exclusivo? É que na COMPUMICRO você tem todas as informações do produto antes mesmo da compra. Ou seja, nossa equipe de analistas, todos de nível superior, estuda o seu caso e indica-lhe a melhor configuração para as suas necessidades. Se você não puder vir ao nosso escritório, onde será recebido com todo conforto e terá à sua disposição um analista com todo o tempo disponível para mostrar-lhe o produto, nós iremos até você. E após a compra continuamos oferecendo nossa assessoria, prestando-lhe assistência técnica, etc...

E sabe quanto você paga a mais por isso? Nada.

Venha comprovar.
Estamos esperando por você.
Pessoalmente.

Venha assistir a uma demonstração do Unitron acessando mais de 300 bancos de dados nos EEUU e França. E mais:

- PROJETO ARIANDA DO SERPRO (TELEMICRO)
- PROJETO CIRANDÃO DA EMBRATEL
- BANCOS DE DADOS PARTICULARES
- TRANSFERÊNCIA TOTAL DE ARQUIVOS ENTRE O UNITRON E IBM-PC COMPATÍVEIS.

compumicro
INFORMÁTICA EMPRESARIAL LTDA.
Rua Sete de Setembro, 99 - 11.º andar
Tel.: PBX (021) 224-7007
CEP 20050 - Rio de Janeiro - RJ

Com a abertura do mundo do teleprocessamento ao usuário de sistemas pessoais, torna-se indispensável saber como se processa a comunicação entre os micros

A viagem dos dados

Roberto Quito de Sant'Anna

Sem dúvida a grande coqueluche do momento – uma vez assentada a poeira causada pela introdução dos micros pessoais no Brasil – é a transmissão de dados ou comunicação entre máquinas. Isto pode ser comprovado pela consolidação do Projeto Ciranda, experiência pioneira da Embratel, pela implantação do Cirandão, da mesma Embratel, do Videotexto da Telesp, e da proliferação dos CBBS (Computer Bulletin Board Systems). Este artigo pretende dar ao leitor uma visão geral e simplificada, tanto quanto o permitir a alta complexidade da tecnologia envolvida, de todo o mecanismo através do qual os dados oriundos do seu micro ou terminal podem atingir o que quer que esteja conectado na outra extremidade da sua linha telefônica.

As redes de comunicação de dados já são usadas há muitos anos nos sistemas de grande porte, tais como os que atendem aos grandes bancos, empresas de aviação e órgãos do Governo, entre outros, sendo que, em termos de computação pessoal foi mais uma vez, o Projeto Ciranda o responsável pelo início de sua difusão entre nós. As vantagens da comunicação de dados são muito numerosas e dentre elas destacamos:

- acesso de um número muito maior de pessoas aos sistemas de Processamento de Dados;
- redução acentuada dos erros de transcrição e de entrada de dados, uma vez que estes são coletados, já em forma legível pela máquina, nos próprios pontos de origem da informação – lojas, postos de gasolina, bancos, etc.;

SERVIÇO	CARACTERÍSTICAS DO SERVIÇO	APLICAÇÕES TÍPICAS	CARACTERÍSTICAS DA COMUNICAÇÃO
INFORMAÇÕES	USUÁRIO RECEBE PERIODICAMENTE POSIÇÕES ATUALIZADAS DE DADOS USA TERMINAIS DE BAIXA VELOCIDADE	INFORMAÇÕES DE CÂMBIO INFORMAÇÕES DE NOTÍCIAS INFORMAÇÕES POLICIAIS	LIGAÇÕES PERIÓDICAS E DE CURTA OU MÉDIA DURAÇÃO
CONSULTA	USUÁRIO PEDE INFORMAÇÕES A UM CENTRO PARA RECEBÊ-LAS LOGO APÓS NÃO INTERAGE COM MEMÓRIA DE DADOS USA TERMINAIS DE BAIXA VELOCIDADE	VERIFICAÇÃO DE CRÉDITO BANCÁRIO INFORMAÇÕES HOSPITALARES INFORMAÇÕES DE TRÂNSITO PESQUISA BIBLIOGRÁFICA	LIGAÇÕES FREQUENTES E DE CURTA DURAÇÃO RESPOSTAS GERALMENTE CURTAS TEMPO DE RESPOSTA CRÍTICO
ATUALIZAÇÃO (ENTRADA DE DADOS)	USUÁRIO FORNECE DADOS DE ATUALIZAÇÃO RESPOSTA NÃO EXIGIDA USA TERMINAIS DE BAIXA VELOCIDADE	CONTROLE DE ESTOQUE DADOS PARA FOLHA DE PAGAMENTO DADOS PARA CONTROLE DE ANDAMENTO DE PROJETOS	INFORMAÇÕES PODEM SER FORNECIDAS A INTERVALOS CURTOS (VÁRIAS VEZES POR DIA) OU MÉDIOS (POR SEMANA) LIGAÇÕES GERALMENTE CURTAS
CONSULTA ATUALIZADA	USUÁRIO FORNECE DADOS E PEDE CONFIRMAÇÃO DE NOVA POSIÇÃO USA TERMINAIS DE BAIXA VELOCIDADE	ATUALIZAÇÃO DE CONTAS (CHEQUE VERIFICADO E LANÇADO) RESERVAS DE VÔO PONTO DE VENDA DE EMPRESAS	LIGAÇÕES FREQUENTES E DE CURTA DURAÇÃO RESPOSTAS GERALMENTE CURTAS TEMPO DE RESPOSTA CRÍTICO
COMPARTILHAMENTO NO TEMPO (TIME SHARING)	ENVIO DE DADOS E PEDIDO DE RESULTADOS USUÁRIO SELECIONA PROGRAMA USA TERMINAIS DE BAIXA VELOCIDADE	SERVIÇOS DE "BUREAU" SOLUÇÃO DE PROBLEMAS GERAIS CÁLCULOS SIMPLES DE PROJETOS DE ENGENHARIA EDIÇÃO DE TEXTOS	LIGAÇÕES FREQUENTES DE VÁRIOS USUÁRIOS LIGAÇÕES DE CURTA E MÉDIA DURAÇÃO TEMPO DE RESPOSTA CRÍTICO
PROCESSAMENTO REMOTO POR LOTES (BATCH PROCESSING)	USUÁRIO ENVIA DADOS E RECEBE RESULTADOS POR LOTES EM OUTRA OCASIÃO USA TERMINAIS DE ALTA VELOCIDADE TEMPO DE PROCESSAMENTO MINUTOS A HORAS	PEDIDO DE CONFECÇÃO DE FOLHAS DE PAGAMENTO EMISSÃO DE ORDENS DE SERVIÇO COM ENTREGA OTIMIZADA (MAIS PRÓXIMA DOS EXECUTANTES)	LIGAÇÕES POUCO FREQUENTES E LONGAS TEMPO DE RESPOSTA NÃO-CRÍTICO
ENTRADA DE PROGRAMA REMOTO (REMOTE JOB ENTRY)	USUÁRIO ENVIA DADOS E RECEBE RESULTADOS POR LOTES USUÁRIO ESCOLHE OU ENVIA PROGRAMAS E PRIORIDADES USA TERMINAIS DE ALTA VELOCIDADE TEMPO DE PROCESSAMENTO ATÉ VÁRIAS HORAS	PROGRAMAS CIENTÍFICOS DE PESQUISA E DE ENGENHARIA	LIGAÇÕES POUCO FREQUENTES E LONGAS TEMPO DE RESPOSTA NÃO-CRÍTICO
COMUNICAÇÃO ENTRE PROCESSADORES	TRANSFERÊNCIA DE GRANDES LOTES DE DADOS E DE PROGRAMAS DE UM PROCESSADOR A OUTRO	DISTRIBUIÇÃO DE CARGA ENTRE COMPUTADORES USO DE BANCOS DE DADOS DISTANTES	LIGAÇÕES POUCO FREQUENTES E COM GRANDE VOLUME TRANSMISSÃO RÁPIDA (ALTA VELOCIDADE)
REQUISITOS COMUNS A TODOS OS SERVIÇOS DE COMUNICAÇÃO DE DADOS: . DISPONIBILIDADE DO SISTEMA, QUANDO SOLICITADO . CONFIABILIDADE NA TRANSMISSÃO . PROTEÇÃO FRENTE A ERROS . SEGURANÇA NA COMUNICAÇÃO			

Figura 1 – Serviços de comunicação de dados. Fonte: BARRADAS, O. e RIBEIRO, Marcelo P., Sistemas analógicos-digitais, *Rio de Janeiro, LTC, 1980, p 989-990.*

- coleta e disseminação imediata da informação, à velocidade eletrônica. Por exemplo, em um banco eletrônico, o saldo da conta do cliente é atualizado instantaneamente após cada transação, ficando imediatamente disponível a todas as agências do país, tornando o cliente um cliente de todo o banco e não de uma única agência;
- redução dos custos operacionais, através de centralização do processamento;
- maior segurança – nos grandes sistemas existem sempre dois ou mais computadores em localizações diferentes, um deles em reserva (stand-by) e em condições de assumir instantaneamente o processamento.

As aplicações da comunicação de dados são, também, muito variadas, e os serviços mais importantes são sumarizados na Figura 1. Para tais aplicações existem dois tipos básicos de ligações a serem estabelecidas: o primeiro, chamado **em-linha** (on-line) é aquele no qual a informação é trocada diretamente com o computador, tipicamente em uma aplicação de consulta realizada por um terminal de caixa bancário ou de balcão de reserva de passagens; o segundo, chamado **fora-de-linha** (off-line), é aquele em que as informações são "estocadas" temporariamente em um dispositivo qualquer de memória para serem posteriormente processadas pelo computador, tipicamente a entrada de programa remoto ou o processamento remoto por lotes.

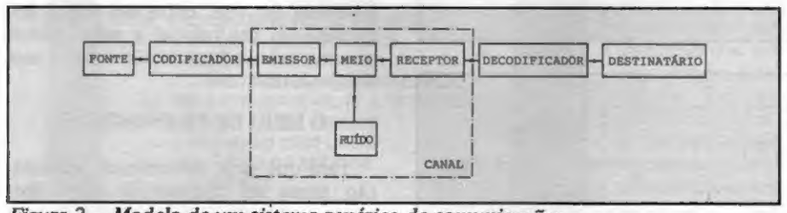


Figura 2 – Modelo de um sistema genérico de comunicações.

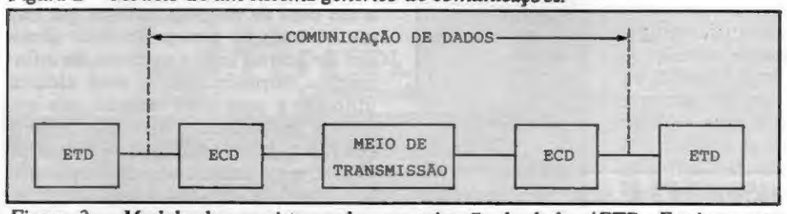


Figura 3 – Modelo de um sistema de comunicação de dados (ETD= Equipamento terminal de dados; ECD=Equipamento de comunicação de dados).

ESTABELECIMENTO DE UM MODELO DE SISTEMA DE COMUNICAÇÕES

Para melhor situar o leitor, nosso passo inicial será estabelecer um modelo que nos permitirá acompanhar todo o processo da comunicação, da origem ou **fonte** ao destino ou **destinatário** (ver Figura 2).

O objetivo de qualquer sistema de comunicações é o transporte da informação ou **mensagem**, em forma tão fiel quanto possível, entre a fonte e o destinatário. Como a fonte e o destinatário podem estar a grande distância um do outro, é necessário que haja um **canal**, encarregado do transporte propriamente dito da mensagem, através de um **meio**, evidentemente com alguma perda de intensidade (**atenuação**), alteração de suas características (**distorção**) e acréscimo de componentes não existentes na mensagem original (**ruído**, representado na figura como uma fonte geradora externa). O **emissor** encarrega-se de colocar a mensagem em uma forma apropriada à transmissão pelo meio, através de um processo chamado **modulação**, além de prover a necessária energia para compensar as perdas durante o trajeto. Por outro lado, o **receptor** retira a energia do meio e recupera a mensagem (demodulação). Como, via de regra, a natureza da informação gerada pela fonte não é adequada ao acionamento do canal, surge a necessidade de mais dois elementos, que completarão o nosso modelo: o **codificador**, que pode dar à mensagem uma forma totalmente diversa, porém a ela inequivocamente relacionada – a letra A, por exemplo, poderia ser transformada no código 11000 – e o **decodificador**, no outro extremo do canal, encarregado de reconstituir a informação.

No caso particular da comunicação de dados, o sistema de comunicações pode ser mais apropriadamente descrito pelo modelo da Figura 3. Nela, os blocos ETD (Equipamento Terminal de Dados) representam a fonte e o destinatário, que podem ser dois computadores ou um terminal e um computador. Os blocos ECD (Equipamento de Comunicação de Dados), por sua vez, representam todo o equipamento necessário à adequação do sinal ao meio de transmissão e vice-versa, realizando as funções do codificador/emissor e do receptor/decodificador.

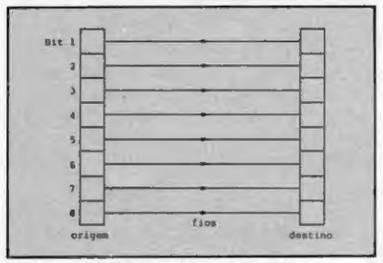


Figura 4 – Transmissão paralela.

TIPOS DE TRANSMISSÃO

Existem dois modos básicos segundo os quais os dados podem ser transmitidos entre dois pontos: o **serial** e o **paralelo**. Imaginemos a transmissão de 1 byte (8 bits) *1 entre um registro de origem e outro de destino (ver Figura 4). Se ligarmos cada um dos bits do registro de origem ao bit correspondente do registro de destino, avisarmos ao registro de destino, de alguma forma, que os dados estão prontos no registro de origem, e permitirmos ao registro de destino aceitar esses dados, teremos uma transferência simultânea de todos os bits, o que caracteriza uma transmissão paralela. Se, por outro lado, tivermos um único fio ligando os dois registros e permitirmos que os bits passem um de cada vez, em seqüência, rumo ao registro de destino, teremos uma transmissão serial. Na Figura 5, os bits 1, 2 e 3 já atingiram o destino, o bit 4 está a caminho, e os bits 5, 6, 7 e 8 aguardam, ainda na origem, a sua vez. Evidentemente, a transmissão paralela é muito mais rápida mas, em compensação, a serial é muito mais barata, por necessitar de apenas uma linha de dados – mais uma vez o eterno compromisso da Engenharia: economizar tempo ou dinheiro? De modo geral, o problema é resolvido assim: no interior do computador, no movimento de dados entre registros da UCP ou entre UCP e memória, onde a velocidade é fator fundamental e as distâncias são curtíssimas, a transmissão é paralela; já a comunicação entre um computador e um terminal é serial, pois, além da economia da interconexão, os dados, mesmo transmitidos serialmente, se deslocam com velocidade muito maior que a de leitura ou de digitação. Resumindo, praticamente toda a transmissão de dados externa ao computador é feita de modo serial. É evidente que, em qualquer caso, todos os caracteres devem ter o mesmo tamanho, ou seja, o mesmo número de bits. Mais adiante falaremos nos códigos usados na transmissão de dados.

A transmissão serial pode ser feita, ainda, de duas formas: **síncrona** e **assíncrona**. Na forma síncrona, os caracteres são transmitidos em um fluxo contínuo, em um único bloco, existindo uma perfeita sincronização entre o emissor e o receptor, de modo que este possa sempre saber o momento exato de "ler" um bit, o início e o término de um caráter e o início e o término da mensagem. O sincronismo pode ser obtido através da transmissão de um trem de pulsos de relógio (clock) em uma linha separada (ver Figura 6) ou dotando-se o receptor de um clock estável, *amarrado* em pulsos de sicronismo transmitidos no início da mensagem. Note que os caracteres são sempre transmitidos, no modo síncrono, sem qualquer intervalo entre eles, o que torna este modo impossível de ser utilizado na ligação entre um terminal e um computador: ninguém pode digitar tão rapidamente. Na transmissão assíncrona, os caracteres podem ser transmitidos aleatoriamente no tempo, com qualquer intervalo entre eles, e sem limitação do tamanho da mensagem. Sempre que for necessário transmitir um caráter, o emissor se encarrega de avisar ao receptor o início da transmissão, através de um bit adicional (start bit = **bit de partida**, correspondente a uma interrupção do sinal na linha) precedendo o código correspondente, e o fim da transmissão, através de um ou dois **bits de parada** (stop bits, correspondendo à condição de marca ou de repouso, isto é, existência de sinal na linha) conforme mostrado na Figura 6. Desta forma, o receptor pode *relaxar*, sabendo que será sempre avisado da transmissão de um caráter com a antecedência suficiente para que possa, através de seu próprio clock, sincronizar seus circuitos para *ler* cada um dos bits no momento apropriado. A transmissão assíncrona tem como principal desvantagem em relação à síncrona uma má utilização do canal. Em compensação, a transmissão síncrona, além de muito mais dispendiosa em termos de equipamento, não pode ser usada em muitos casos, como o mostrado acima para o terminal. Na ligação que mais nos interessa, ou seja, entre um micro doméstico ou profissional e outro micro ou uma rede, a transmissão sempre será serial e assíncrona.

1. Nota do autor – Um bit é igual a um dígito binário, isto é, a menor unidade de informação existente em um sistema de computação: pode assumir, a cada instante, apenas um entre dois valores possíveis, 0 e 1.

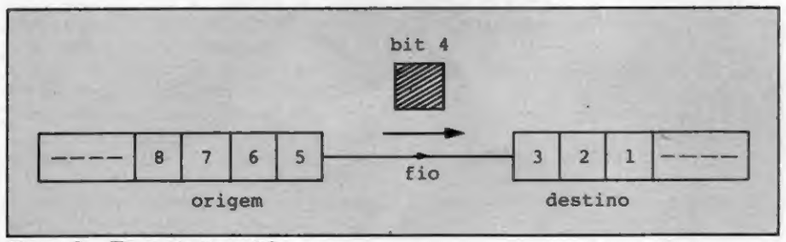


Figura 5 – Transmissão serial.

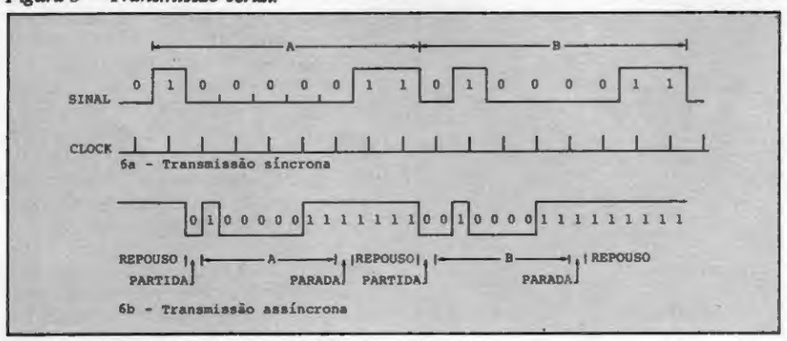


Figura 6 – Exemplos de transmissão dos caracteres A e B no modo síncrono (6a) e no modo assíncrono (6b), com 1 bit de parada e paridade ímpar. Note que as escalas são diferentes.

O MEIO DE TRANSMISSÃO

Para que uma determinada informação possa ser transmitida entre dois pontos, a mesma tem que ser superposta a um sinal de natureza elétrica, que terá um ou mais de seus parâmetros alterados de acordo com a natureza da informação. Normalmente o sinal elétrico utilizado é uma onda senoidal cuja amplitude instantânea é dada por $x(t) = A \cos(2 \pi ft + \theta)$, onde t é o tempo em segundos e A (amplitude), f (freqüência) e θ (fase) são os parâmetros que podemos fazer variar. Se variarmos o parâmetro desejado de forma contínua, de modo a constituir uma réplica da informação original, o sinal resultante será dito um sinal **analógico**; se, por outro lado, permitirmos que o parâmetro a ser variado assuma somente um certo número de valores, chamados de valores ou níveis discretos, estaremos em presença de um sinal **digital**. O caso mais conhecido de sinal digital, aquele que possui apenas dois níveis, é o sinal binário. Os sinais podem, ainda, ser submetidos a processos de codificação, com o resultado final diferindo completamente do sinal inicial; o importante é que o conteúdo da informação se mantém inalterado e pode ser integralmente reconstituído no destino.

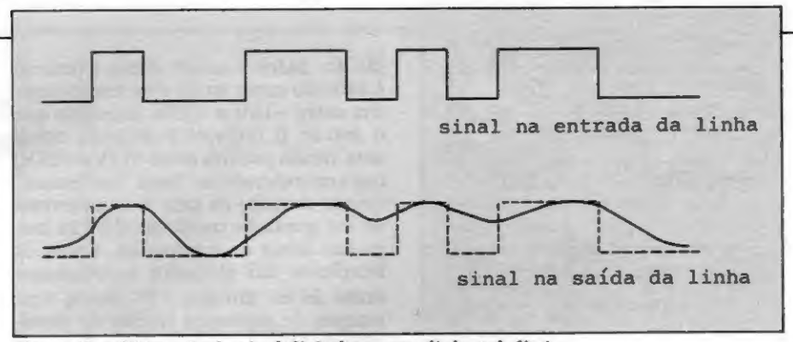


Figura 7 – Distorção do sinal digital em uma linha telefônica.

O processo segundo o qual alteramos um ou mais dos parâmetros de um sinal é chamado **modulação**, e o sinal modificado, que vai transportar a nossa informação até o destino, é chamado de **onda portadora**.

O meio de transmissão por excelência para a transmissão de dados é o canal telefônico comum, acessível através de um par de fios de nossa linha telefônica, projetada e instalada para a transmissão de voz em forma analógica.

A voz humana é um sinal complexo e a sua energia está distribuída de modo não uniforme em uma faixa de freqüências compreendida entre 15Hz e 15000 Hz, aproximadamente, com a maior concentração ocorrendo entre 300Hz e 3400Hz. Por questões de economia, os canais de voz transmitem apenas essa faixa de freqüências, chamada de banda passante, largura de banda ou largura de faixa da linha ($B = f_2 - f_1 = 3100$ Hz). A banda passante é a principal característica de um canal de voz, sendo a responsável pela velocidade máxima de transmissão, em bits por segundo (bps), do canal. Os canais telefônicos podem ser **comutados** (o destino é atingido através de uma rota escolhida ao acaso, em função das disponibilidades da rede telefônica, como em uma ligação comum), ou **privativos** (dedicados, alugados), constituindo uma ligação ponto-a-ponto, disponível ao usuário 24 horas por dia. A escolha entre comutada e privativa depende de uma série de fatores, principalmente do volume de tráfego e, como regra geral, a linha privativa oferece melhor qualidade de transmissão. Embora teoricamente muito maiores, as velocidades máximas de transmissão obtidas em linhas telefônicas ficam, na prática, limitadas a 9600 bps, em virtude de outras características restritivas, tais como a atenuação, distorção, ruído, eco e estabilidade. O leitor mais curioso no assunto poderá *queimar pestanas* durante muitas horas consultando a bibliografia citada.

O EMISSOR E O RECEPTOR

Já vimos que as linhas telefônicas foram projetadas para transmitir freqüências de voz na faixa de 300-3400 Hz, e que os sinais de voz são sinais analógicos. Se injetarmos em uma linha telefônica os sinais binários oriundos de nosso computador, o resultado na outra extremidade será o mostrado na Figura 7: ao invés de um sinal claro, de transições bem nítidas, obteremos um sinal distorcido, no qual as transições se mostram bem atenuadas, e que poderá ser mal interpretado pelo equipamento de recepção, que terá eventualmente dificuldade de distinguir entre os níveis 0 e 1. A distorção será tanto maior quanto mais estreita for a banda passante da linha, pois a decomposição de um sinal binário nos mostra que nele estão presentes componentes de altíssima freqüência, as quais serão brutalmente atenuadas ao passarem pela linha, sendo virtualmente inexistentes na saída. Como seria economicamente inviável aumentar a largura de banda das linhas telefônicas (elas chegaram primeiro, lembre-se), e já que elas atendem perfeitamente à finalidade para a qual foram projetadas, a solução mais inteligente e que foi a adotada é a de adaptar o sinal à linha, o que pode ser feito através de um **modem**. O modem, cujo nome é formado pela contração das palavras modulador e demodulador, é um equipamento bidirecional que, instalado nas duas extremidades de um canal de comunicação de dados, tem por função adequar um sinal binário oriundo de um computador às características da linha (funcionando como emissor), e vice-versa (funcionando como receptor). Para a maioria dos efeitos práticos, o modem é o próprio ECD da Figura 3.

O tipo mais comum de modem é o chamado **modem analógico**, através do qual os níveis binários 1 e 0 (também chamados de **marca** e **espaço**, respectivamente) são transformados em tons senoidais puros, que vão modular uma portadora senoidal cuja freqüência está dentro da banda passante da linha telefônica, podendo, então, ser transmitida praticamente sem distorção. Na extremidade de destino, um outro modem se encarrega de demodular esta portadora, extraindo da mesma os tons de marca e de espaço, que, após reconvertidos em níveis binários, serão entregues ao computador (ver Figura 8). Simples, não?

Dado o caráter universal das redes de telecomunicações, torna-se necessária uma normalização ou padronização rigorosa dos equipamentos. Assim, a União Internacional de Telecomunicações (UIT), da qual o Brasil é membro, através de seu Comitê Consultivo Internacional de Telegrafia e Telefonia (CCITT) estabeleceu o chamado **padrão CCITT** de modems, também conhecido como **padrão europeu**, adotado pelo Brasil. Outros países, liderados pelos Estados Unidos, utilizam o chamado **padrão BELL**, ou **padrão americano**, normalizado pelo Bell System.

As normas estabelecidas pelo CCITT dizem respeito, basicamente, às taxas (ou velocidades) de transmissão da informação, sendo as mais usuais as de 150, 300, 600, 1200, 2400, 4800 e 9600 bps, e aos tipos de modulação, normalmente em freqüência (FSK = Frequency Shift Keying = modulação por desvio de freqüência) ou em fase (PSK = Phase Shift Keying = modulação por

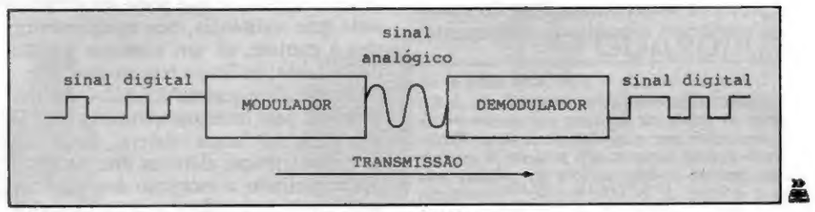


Figura 8 – Modulação e demodulação do sinal digital.

desvio de fase), este para velocidades acima de 1200 bps. Ainda um mesmo modem pode ter velocidades diferentes para transmissão e recepção: o tipo utilizado para acesso ao Videotexto transmite a 75 bps e recebe a 1200 bps *2. No campo da computação pessoal, contudo, a tendência é adotar a comunicação serial, assíncrona, a 300 bps. Existe uma certa tendência em confundir bps e baud como unidades de medida de velocidade de transmissão. A unidade baud, que recebe este nome em homenagem a Baudot, um dos pioneiros das telecomunicações, é mais corretamente aplicada à medida de velocidade de transmissão de sinais telegráficos. Baud representa o número de vezes que o estado da linha se modifica por segundo. Como, na maioria das aplicações de teleprocessamento, a condição da linha é alterada exatamente pela presença ou ausência de sinal, o número que mede a velocidade em bps é o mesmo que a que mede em baud, daí a confusão. Por via das dúvidas, a melhor maneira de nunca errar é expressar a velocidade sempre em bps.

Existe ainda um tipo de modem, conhecido como modem digital. A rigor este tipo não deveria ser chamado de modem, uma vez que não realiza a modulação/demodulação do sinal, e sim uma simples mudança na sua representação digital (codificação) e na representação elétrica (forma do sinal), transformando-o em um outro sinal digital, porém mais adequado às condições da linha. Embora seu alcance seja muito restrito, não ultrapassando 300m, constitui uma solução econômica e aceitável para, por exemplo, ligações dentro de um mesmo prédio. Os modems digitais não são normalizados pelo CCITT, não havendo, portanto, compatibilidade entre os modelos dos diversos fabricantes.

O alcance dos modems digitais diminui conforme aumenta a velocidade de transmissão. Transmitindo a 300 bps, pode-se operar com um modem digital em distância de até 4.500m. Já a 600 bps, o alcance deste equipamento diminui para 300 metros. Por suas características, os modems digitais em geral são bem mais baratos que os analógicos.

Outro equipamento não padronizado pelo CCITT é o **acoplador acústico**, dotado de um bocal emissor e outro receptor nos quais o monofone do aparelho telefônico é encaixado diretamente,

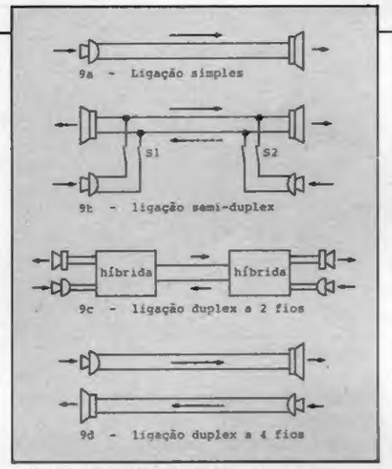


Figura 9 – Modos de ligação.

com toda a transferência de informação ocorrendo *pelo ar*, sem qualquer ligação elétrica com a rede telefônica. Além de sujeitos a interferências de ruídos externos, causados muitas vezes pelo fato do fone do aparelho telefônico não se adaptar ao acoplador, sua velocidade de transmissão é relativamente baixa, não ultrapassando 300 bps.

A ligação entre os modems pode ser feita ainda de três modos (ver Figura 9): o modo **simplex**, sem utilidade prática, pois permite a comunicação apenas em um único sentido; o modo **semi-duplex** (half-duplex), que permite a ligação nos dois sentidos, porém não simultaneamente; e finalmente, o modo **duplex** (full-duplex), permitindo a comunicação simultânea nos dois sentidos, e que pode ser a dois fios (exatamente como o telefone a que estamos habituados, utilizando um circuito híbrido ou um acoplador direcional para separar os sinais emitido e recebido) ou a quatro fios, mais confiável porém mais caro, por necessitar de duas linhas separadas.

O CODIFICADOR E O DECODIFICADOR

Em nosso sistema de comunicação de dados as funções de codificação e de decodificação são normalizadas pela EIA – Electronic Industries Association, dos EUA, através do Padrão RS 232-C, compatível com o CCITT, cuja realização física é conhecida como **interface padrão** RS 232-C (o "C" significa a terceira versão) e que pode ser identificada pela existência, nos equipamentos que a contém, de um conector padrão de 25 pinos, de forma trapezoidal.

A interface padrão RS 232-C é a responsável pela interconexão entre o ETD e o ECD, em forma bilateral, definindo as características elétricas dos circuitos de transmissão e recepção de dados, os seus níveis de tensão e os sinais de dados e de controle necessários. Na transmissão de dados, o estado lógico 1 (marca) é definido como sendo uma tensão negativa entre –15V e –25V, enquanto que o estado 0 (espaço) é definido como uma tensão positiva entre +15V e +25V, tudo referenciado ao "terra" ou "massa" (ponto comum) de sinal e com previsão de um queda de tensão de ±12V ao longo das linhas de transmissão. Como os receptores são obrigados a reconhecer sinais de no mínimo ±3V, sobra uma margem de segurança (região de transição) de 6V entre os níveis 1 e 0, o que contribui para aumentar a imunidade a ruídos e a diferenças de potencial de massa.

OS CÓDIGOS

Um dos mais importantes passos para o desenvolvimento da comunicação de dados foi a padronização dos códigos, visando a que os diversos equipamentos pudessem "falar" entre si. O primeiro esforço de padronização data de 1963, através do código ASCII63 (ASCII é a sigla de American Standard Code for Information Interchange –, Código Padrão Americano para Intercâmbio de Informação), e a versão atual do código ASCII, surgida em 1968, adotada em âmbito mundial. Ver Figura 10.

O código ASCII é um código de de 7 bits, possibilitando um total de 128 (= 2^7) combinações válidas. A esses 7 bits é adicionado um oitavo bit, chamado **bit de paridade**, com o objetivo de diminuir a incidência de erros na transmissão. Por exemplo, o bit de paridade poderá ser 0 ou 1 conforme o número de bits 1 do código considerado seja par ou ímpar – o receptor conta os bits 1 de cada código e, caso a contagem não seja um número par (paridade par), envia um sinal ao emissor para que este transmita novamente o código. É evidente que se, devido ao ruído, houver a inversão de dois bits quaisquer, o erro não poderá ser detectado por este método. Os bits adicionais introduzidos nos códigos, como o bit de paridade, não contém informação, sendo chamados de **redundantes**. Quanto maior for a redundância de um código, menor será a **eficiência** do canal, definida como o resultado da divisão do número de bits de informação (os bits *úteis*) pelo número total de bits transmitidos.

Outros códigos normalmente usados em comunicação de dados são o **Baudot** (para teleimpressores) e o EBCDIC (Extended Binary Coded Decimal Interchange Code), usado nos equipamentos IBM.

Além da mensagem propriamente dita, deve transitar pelo canal um constante fluxo de informações entre as máquinas envolvidas na comunicação. Esse fluxo de informações, que é o responsável pelo estabelecimento, manutenção, controle e desconexão da comunicação, recebe o nome de **protocolo** (handshaking, "aperto de mãos"). As duas colunas da esquerda da Figura 10 mostram os caracteres de controle do código ASCII, e seus significados constam da legenda. Remeto o leitor interessado, mais uma vez, à bibliografia especializada, para maior aprofundamento no assunto.

CÓDIGO: $b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1$

CONFORME OS BITS DA TABELA

BITS $b_4\ b_3\ b_2\ b_1$	BITS $b_7\ b_6\ b_5$ 000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	8	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	_	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

CONTROLE — INFORMAÇÃO

LEGENDA:

NUL = all zeros	VT = vertical tabulation	SYN = synchronous/IDLE
SOH = start of heading	FF = form feed	ETB = end of transmitted block
STX = start of text	CR = carriage return	CAN = cancel (error in data)
ETX = end of text	SO = shift out	EM = end of medium
EOT = end of transmission	SI = shift in	SUB = start of special sequence
ENQ = enquiry	DLE = data link escape	ESC = escape
ACK = acknowledgement	DC 1 = devide control 1	FS = information file separator
BEL = bell or attention signal	DC 2 = devide control 2	GS = information group separator
BS = back space	DC 3 = devide control 3	RS = information record separator
HT = horizontal tabulation	DC 4 = devide control 4	US = information unit separator
LF = line feed	NAK = negative acknowledgement	DEL = delete

Figura 10 – Código ASCII. Fonte: BARRADAS, O e RIBEIRO, Marcelo P., Sistemas analógicos-digitais. *Rio de Janeiro, LTC, 1980, p 1049.*

CONCLUSÃO

Espero que este artigo tenha conseguido satisfazer a curiosidade do leitor apenas curioso e que tenha fornecido àquele mais interessado, desejoso de maiores conhecimentos, o embasamento necessário à leitura dos *papiros* especializados no assunto.

Para finalizar, um lembrete: de nada adianta toda a parafernália de equipamentos e técnicas de comunicação de dados se não dispusermos do **software de comunicação**, indispensável ao gerenciamento de todo o processo e, por si só, assunto para muitas e muitas páginas. Por uma questão de fidelidade ao objetivo, que foi o de abordar apenas os aspectos técnicos da comunicação, a sua não citação no texto foi intencional. De qualquer modo, aqui, como em qualquer outra aplicação, é o software que torna o computador em algo útil – sem ele, o nosso computador não passará de um enfeite (?) de mesa ou mero peso de papel...

BIBLIOGRAFIA

BARRADAS, O. e RIBEIRO, Marcelo P., *Sistemas analógicos-digitais.* Rio de Janeiro, LTC, 1980.

COUGER, J. Daniel & McFADDEN, Fred R., *First course in data processing with* BASIC. USA, John Wiley & Sons, 1981.

EMBRATEL, *Básico de comunicação de dados,* edição experimental. Rio de Janeiro, DTR/EMBRATEL, 1984.

McNAMARA, J. E., *Technical aspects of data communication.* USA, Digital Equipment Corporation, 1977.

PEREIRA FILHO, Jorge da C. et al., *Equipamentos e sistemas de computação,* Coleção Computadores para Usuários, Vol. 2. Rio de Janeiro, Campus, 1984.

TAROUCO, Liane M., *Redes de comunicação de dados.* Rio de Janeiro, LTC, 1977.

Roberto Quito de Sant'Anna é Engenheiro de Telecomunicações, formado pelo Instituto Militar de Engenharia e Professor da cadeira de Informática da Academia Militar das Agulhas Negras.

** 2. Nota da redação – A variação entre as velocidades de recepção e emissão de dados pode ser explicada uma vez que o número de informações que o usuário do sistema Videotexto deverá fornecer são poucas, já que ele basicamente apenas escolhe as páginas que quer acessar. Já as informações provenientes do banco de dados são muitas, o que requer uma velocidade maior na transmissão.*

Eis as diferenças e algumas vantagens em se conjugar o verbo compilar, ao invés de interpretar, em se tratando de linguagem BASIC

BASIC interpretado x compilado

Marcelo Renato Rodrigues

O sistema completo de programação BASIC deve traduzir as suas instruções BASIC em instruções que o microcomputador entenda, ou seja, código-objeto. Os meios empregados para fazer essa conversão dependem do sistema BASIC que você tem disponível, normalmente o interpretador BASIC.

O interpretador converte cada instrução para o código-objeto, executando-a imediatamente após a conversão. Isso é feito toda a vez em que o programa é rodado. O compilador, por outro lado, converte todo o programa em código-objeto. Então, você terá o seu programa sob duas formas: o programa-fonte, em BASIC, e o programa-objeto, em linguagem de máquina. Este último, quando submetido, dispensará a conversão das instruções, atividade do interpretador. Para melhor entendimento desta análise, consideremos o BASIC da linha TRS 80, modelo III e o compilador BASIC da Radio Shack, o RSBASIC.

VANTAGENS DO COMPILADOR

O RSBASIC traduz o programa-fonte numa linguagem intermediária, isto é, entre o BASIC e a linguagem de máquina. Entre as vantagens enumeradas pelo fabricante duas merecem atenção: só o autor do programa poderá conhecê-lo, pois é o único dono do programa-fonte e a linguagem intermediária é desconhecida; além de sua economia de memória e espaço em disco.

A primeira tem importância para o programador que pretende comercializar os seus aplicativos. A segunda vantagem é ofuscada pelo grande espaço que o compilador ocupa na memória. Mas a vantagem é absolutamente verdadeira com relação aos arquivos em disco.

O usuário do compilador será inicialmente surpreendido, tanto pelo maior rigor sintático das instruções – por exemplo, observância dos espaços entre as palavras componentes das instruções –, quanto pelos produtos documentais do processo de compilação, como a listagem comentada do programa, o mapa das variáveis e a listagem de referência cruzada, na qual são relacionadas as variáveis e as linhas do programa-fonte onde elas são referenciadas (figura 1). Tais produtos são familiares ao usuário que trabalhou ou trabalha com computadores de maior porte.

DIFERENÇAS DE LINGUAGEM

Um aspecto importante a ser demonstrado é quanto às diferenças de linguagem dos dois processos. A primeira delas é com relação à maior precisão do BASIC compilado quanto à alocação de espaço na RAM, inexistindo a instrução **CLEAR n**, que executa a alocação global de espaço para strings. Assim, a reserva de espaço é feita variável a variável, através das instruções **DIM** ou **STRING**. Não havendo essa descrição, o compilador considerará o *default* de 255 bytes por variável.

Ainda com relação à definição de variáveis, outra diferença é o número de dígitos para o nome da variável, que passa de três para seis, permitindo ter, por exemplo, duas variáveis distintas – SALDO1 e SALDO2 – impossível no interpretador, que consideraria para os dois casos apenas a variável SAL. E entre as instruções que atribuem valores às variáveis, há três diferenças significativas.

A primeira delas é uma variação do **RESTORE**, que permite apontar a seqüência DATA a partir da qual nos interessa restaurar, através do apontamento do número da linha que a contém. Não se fornecendo o número da linha, a instrução funciona exatamente da forma usual.

A segunda é a instrução **SWAP**, que troca valores entre duas variáveis, muito empregada em reordenações. A terceira e última, a instrução **INPUT**, embora continue sendo de uso incomodo, foi aperfeiçoada com formatação dos dados de entrada e especificação do número de dígitos da variável.

SEGMENTAÇÃO DE PROGRAMAS

Entre os dois sistemas, existem diferenças significativas, que certamente *farão a cabeça* de usuários mais exigentes. Por exemplo, são disponíveis dois recursos poderosos voltados à segmentação de programas durante a execução: a transferência de controle para subprogramas e encadeamento de programas.

Subprogramas são sub-rotinas mais potentes que as usuais, pois trabalham com dados armazenados sob diferentes nomes de variáveis. Assim como as sub-rotinas comuns, os subprogramas são chamados pelo programa principal e, após sua execução, retornam a ele. O exercício de sua aplicação revela as seguintes vantagens em relação à sub-rotina convencional:

```
RSBASIC  ver 2.3          EXEMPLO/BAS
09/11/84                19:18:17              PAGE 1
0000      00010 REM *PROGRAMA EXEMPLO COM UTILIZACAO*
0000      00020 REM *        DO COMPILER BASIC       *
0000      00030 REM *  O PROGRAMA SOLICITA NOME      *
0000      00040 REM *        E SOBRENOME             *
0000      00050 DIM SOBRE$20
0000      00060 PRINT "QUAL E' SEU SOBRENOME?"
000F      00070 PRINT CRT(2,0);: INPUT SOBRE$
002D      00080 PRINT CRT(6,0); "QUAL E' O SEU NOME?"
0040      00090 PRINT CRT(2,0);: INPUT NOME$
005E      00100 PRINT CRT(12,0); "OBRIGADO, "; NOME$; " "; SOBRE$; "!"
0079      00110 END
SYMBOLIC MEMORY MAP
SCALARS
00D6    NOME     STRING*255       00DD    SOBRE    STRING*20
CROSS REFERENCE LISTING
SCALARS
NOME                      90      100
SOBRE                     50       70      100
FINAL SUMMARY
  245 (00F5) BYTES OF PROGRAM
  278 (0116) BYTES OF LOCAL DATA
   11 SOURCE LINES
   13 SOURCE STATEMENTS
*** COMPILATION COMPLETE ***
```

Figura 1

- O subprograma não é chamado pelo número da linha, mas pelo nome;
- Os dados transferidos ao subprograma não necessitam de adequaçao quanto ao nome das variáveis; basta apenas existir compatibilidade entre elas, pois o mesmo dado terá um nome no programa principal e outro no subprograma;
- Pode-se transferir matrizes ao subprograma.

O subprograma é compilado com o programa principal, sendo integrante dele, disputando espaço na RAM, mas adicionando incrível flexibilidade a seus programas.

Já o encadeamento de programas (CHAIN) executa a segmentação sem a ocupação simultânea de espaço na RAM. É um método de dividir um programa muito grande em outros menores e menos complexos, sendo cada um deles carregado na memória e executado separadamente, embora trocando dados comuns.

ENTRADA/SAÍDA PARA TECLADO E MONITOR DE VÍDEO

A formatação de dados para entrada/saída é um grande avanço e a saída via monitor tem duas funçoes especiais para posicionamento do cursor. A função **CRT** move o cursor para uma específica locação linha coluna e a função **CRTR (x,y)** move o cursor x linhas e y colunas, a partir da posição atual. É o adeus ao **PRINT @**.

São acrescidas, ainda, funções para localização da posição do cursor **CRTy** e **CRTx**, que fazem retornar os valores da linha-coluna aonde se encontra o cursor, e uma função para leitura de área especificada no vídeo.

ENTRADA/SAÍDA PARA ARQUIVOS EM DISCO

Diferenças importantes são observadas na manipulação de arquivos em disco. Além de criar arquivos seqüenciais (figura 2) e randômicos ou diretos (figura 3), o RSBASIC elabora o arquivo ISAM (Indexed Sequential Acess Method), isto é, o arquivo seqüencial indexado utilizado pelos sistemas maiores, nos quais os registros são alcançados por chaves de acesso e não pelo número de registro. Por exemplo, num arquivo de nomes e endereços, a chave de acesso pode ser o sobrenome. Na leitura, os registros são apresentados segundo a classificação, em ordem alfabética, da chave de acesso, como no exemplo da figura 4, no qual a chave de acesso é a primeira letra do nome.

De uma forma geral, a entrada/saída de arquivos pode ser string ou numérica, não havendo necessidade de converter dados numéricos em strings para gravar, e vice-versa, após a leitura. A inexistência da instrução **FIELD** facilita também a leitura-gravação de vetores. Para entrada/saída há três métodos:

- Seriado – as vírgulas separam os campos dos registros;
- Formatado – é empregada imagem-padrao para controlar a disposição dos campos;
- Binário – os dados numéricos são arquivados exatamente como estão na memória.

Além dessas, há outras diferenças mais ou menos sutis em instruções, funções e comandos que, se expostos, levariam a um tratamento mais aprofundado. Com relação às facilidades de grande interesse, há o **RUNTIME**, subsistema que apenas roda programas, ocupando menor espaço na memória; o **BEDIT**, eficiente editor **BASIC**; e o **DEBUG**, depurador de programas.

Como se vê, há numerosas vantagens com relação aos recursos de linguagem, tornando o BASIC bem mais potente. Mas como no Brasil o emprego do BASIC compilado ainda é restrito, surgem problemas de disponibilidade de aplicativos no mercado e de incompatibilidades, pois programas estruturados e desenvolvidos em BASIC compilado não são compatíveis com o BASIC interpretado e vice-versa.

Marcelo Renato Rodrigues é engenheiro eletricista formado pela Escola Politécnica da Universidade de São Paulo, em 1968. Trabalha na Companhia Energética de São Paulo (CESP) como Assessor de Planejamento da Vice-Presidência de Produção de Transmissão de Energia Elétrica.

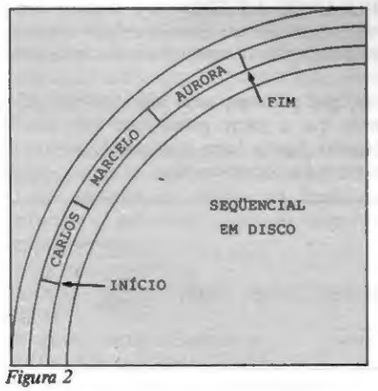


Figura 2

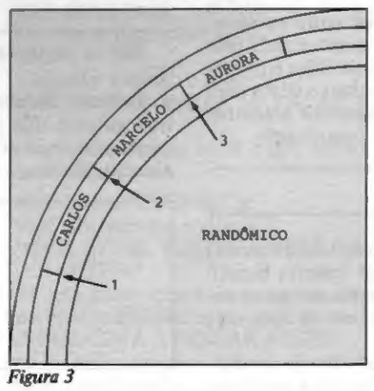


Figura 3

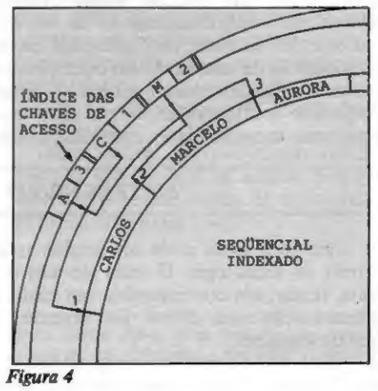


Figura 4

Compartilhar recursos de custos elevados e informações produzidas por diversas estações são algumas vantagens obtidas com o uso de redes locais

Redes locais

Amaury Moraes Junior

Uma rede de computadores consiste em um certo número de computadores interligados por um sistema de comunicação. Dentro dessa filosofia, surgiu mais recentemente um tipo particular de rede, chamado Rede Local (Local Area Network – LAN). Nessas redes, as principais características são a extensão geográfica, de no máximo poucos quilômetros; a alta taxa de transmissão, de 5 a 10 Mb/segundo; e a ausência de um processador central, isto é, todos os elementos conectados à rede possuem capacidade de processamento.

Na medida em que as organizações (bancos, indústrias, hospitais etc.) começaram a possuir um maior número de computadores, principalmente com o advento dos computadores de baixo custo, tornou-se necessário que estes equipamentos se interconectassem, para compartilhar recursos e informações.

É importante observar que embora o custo dos computadores tenha diminuído constantemente, o preço dos equipamentos periféricos (discos, impressoras etc.) não acompanharam esta redução, e seu alto custo justifica o seu compartilhamento entre vários usuários.

Entre as vantagens na utilização de uma rede local, podemos citar o aumento dos recursos físicos (periféricos) disponíveis para cada estação; maior integração entre aplicações, através do compartilhamento de informações entre as diversas estações da rede; confiabilidade elevada, caracterizada pela inexistência de um elemento centralizador (cuja falha comprometeria o funcionamento global do sistema); baixo custo para pequenas configurações; e, por fim, o crescimento gradativo conforme as necessidades computacionais da organização.

EM TRÊS NÍVEIS

Uma rede local pode ser implementada em três diferentes níveis de tecnologia. O mais elevado oferece maiores benefícios, tendo, em contrapartida um custo também elevado e implementação mais difícil. Em resumo, os níveis de uma rede são os seguintes:

- Nível 1 – O objetivo neste estágio é o de que vários usuários possam compartilhar periféricos como impressoras, plotters, modems, equipamentos geralmente de preços elevados e que, utilizados por mais de uma estação têm seu preço real dividido pelos departamentos. E devido ao baixo volume de saída desses dispositivos, não há degradação do sistema.
- Nível 2 – Os meios de armazenamento de massa, normalmente discos do tipo Winchester são compartilhados pelos diversos usuários do sistema. Porém, é necessário que a rede local tenha capacidade de transmitir dados em altas velocidades. O compartilhamento requer software de controle de acesso a esses arquivos, para que a integridade das informações seja mantida.
- Nível 3 – Neste nível não se trata de compartilhar dispositivos físicos, mas sim a informação existente no ambiente da rede local. Além da capacidade de transmitir em altas velocidades, este nível requer facilidade no acesso simultâneo a arquivos e possibilidade de bloquear registros (lock), todos importantes para que se possa compartilhar informações. Integridade dos dados, processamento distribuído, eliminação de redundância de informações, possibilidade de consolidar dados produzidos por diferentes pessoas são algumas das vantagens da implementação deste nível em rede local.

FORMANDO A REDE

São os seguintes os principais componentes de uma rede (Figura 1):

- **Unidade de interface**, que pode ser uma placa ou um gabinete externo. Ela permite que o micro possa *falar* com a rede local, e as mais sofisticadas fazem todo o processo de comunicação, liberando o micro para outras tarefas.

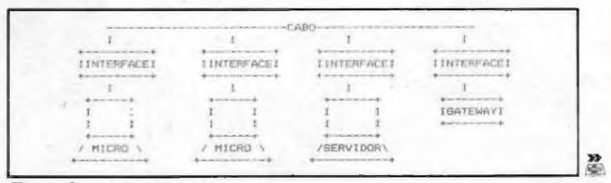


Figura 1

Tem hora que precisa ser micro.

O Elppa II Plus é um micro computador.
Só que tem macro vantagens.
É feito quase artezanalmente, portanto testado um a um.
E isso é uma macro qualidade.
Como é feito com componentes de alta qualidade, dentro dos melhores padrões de Engenharia, a confiabilidade do Elppa II Plus é macro.
O custo de manutenção é micro: o único com um ano de garantia - macro qualidade com macro garantia.
Já com o preço acontece uma coisa interessante, deveria ser macro, mas quando você verifica o custo de uma configuração vê que é micro.
A assistência técnica é macro - direta do fabricante ou através de seus credenciados.
Ele é um Apple® compatível e dispõe de vasta gama de expansões e periféricos à sua disposição - CONTROLADOR DE DRIVE, CP/M, PAL-M, 80 COLUNAS, SOFTSWITCH, 16K, 64K, 128K, GRAPH +, SUPER SERIAL CARD, SINTETIZADOR DE VOZ, MONITOR III, etc... - **macro vantagem.**

Tem hora que precisa ser macro.

Conclusão: Seja para você ou para sua empresa, micro ou macro, faça como a Rede Globo, a Rede Bandeirantes ou a Control que têm se utilizado do Elppa II Plus em suas necessidades empresariais ou como os funcionários do Bamerindus para suas atividades profissionais e de lazer.
Faça como tantos outros, que estão aproveitando as vantagens de um micro que sabe ser macro na hora certa.

Macro garantia 1 ano inteirinho.

Escolha o Elppa II Plus a macro escolha.

elppa II plus

O micro macro.

Victor

Fábrica: Rua Aimbere nº 931 - S.P. Tel. 864.0979 - 872.2134
Show Room: Av. Sumaré nº 1.744 - S.P. Tel. 872.4788

• **São Paulo** - Audio 282-3377 - ADP System 227-4433 - Bruno Blois 223-7011 - BMK 62-9120 - Europlan 256-9188 - Victor Show Room 872-4788 • **Rio de Janeiro** - CML 285-6397 - Eleceeme 201-3792 - Formed 266-4722 - Sistema 253-0645 - SC Sistemas 232-8304 • **Belo Horizonte** - Spress 225-8988 • **Porto Alegre** - Aplitec 24-0465 - DB Computadores 22-5136 - Embramic 41-9760 • **Vitória** - Metaldata 225-4700 - Soft Center 223-5147 • **Brasília** - Compushow 273-2128 • **Curitiba** - Video e Audio 234-0888 • **Londrina** - Set In 23-6183 • **Recife** - NC Sistemas 228-0160 — Tecromic 325-3363 • **Florianópolis** - Micro Home 23-2283 • **São José do Rio Preto** - Teledata 33-2714 • **Fortaleza** - Systematic 244-4746

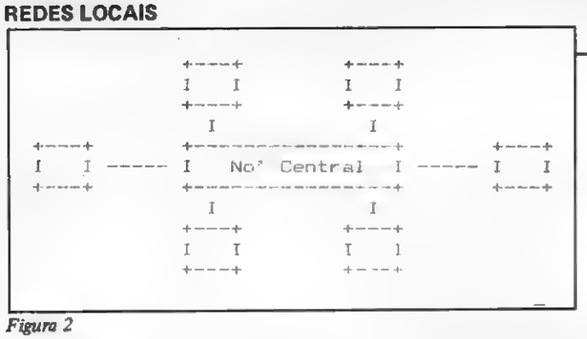


Figura 2

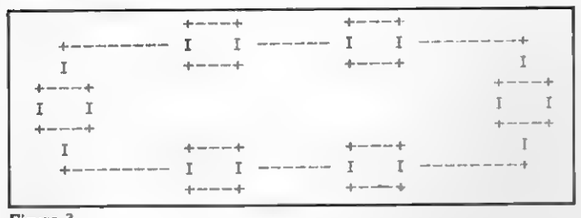


Figura 3

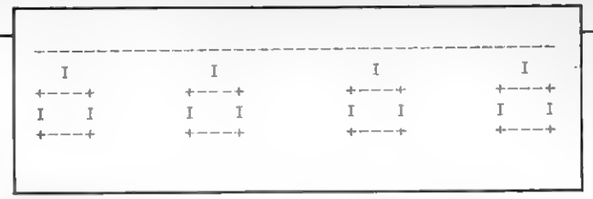


Figura 4

• **Cabo físico** para conectar as estações à rede local. Os tipos comumente usados são o par trançado, para pequenas distâncias (até 300 metros) e o cabo coaxial de custo mais elevado, usado para grandes distâncias. Com transmissão em banda base este último atinge 500 metros, enquanto que com a transmissão em banda larga pode atingir até 50 quilômetros.

• **Servidores**, que normalmente gerenciam o compartilhamento de arquivos ou impressoras, podem estar residentes em uma estação de trabalho ou em uma unidade dedicada para esse fim.

• **Gateways** são computadores, dedicados ou não, que permitem às estações a possibilidade de comunicação com outras redes e serviços externos ao âmbito da rede local.

Topologia é a forma física de interconexão dos elementos da rede. Há três tipos básicos para as redes locais.

Na topologia em estrela (Figura 2) todos os nós (ou estações) são ligados a um nó central, através do qual os dados passam. Neste tipo é comum o nó central possuir maior capacidade de processamento, além de concentrar os periféricos que são compartilhados entre as outras estações. A rede em estrela apresenta sua maior deficiência na confiabilidade, qualquer falha no nó central causa a parada total do sistema, além de ser limitada em termos de expansão, normalmente a oito estações. Seu desempenho também é determinado pela capacidade de processamento do nó central.

Uma rede organizada em anel é composta de estações ligadas em série (Figura 3), formando uma espécie de círculo. Normalmente, cada estação é ligada à rede através de uma interface especial, cuja responsabilidade é retransmitir os dados que não se destinam àquela estação, ler os dados destinados ao nó e inserir dados. Devido ao fato de as redes em anel exigirem uma interface ativa para seu funcionamento, a confiabilidade da rede se reduz à confiabilidade das interfaces. A falha de qualquer uma delas seccionará o sistema. Na topologia em anel também podem surgir problemas relacionados com falhas ou erros no processamento de mensagens. Por outro lado, pode crescer ilimitadamente. Contudo é importante lembrar que cada interface introduzida no sistema provocará um retardo adicional na rede e a degradação pode se tornar indesejável, se muitas interfaces estiverem presentes na rede.

Na topologia em barra comum (BUS) os nós compartilham o meio de transmissão através de interfaces passivas, isto é, o funcionamento da rede não depende do funcionamento das interfaces. Uma vez que a barra é compartilhada por todos os nós (Figura 4), o acesso a ela deve ser controlado, de forma centralizada ou distribuída. No caso centralizado, a mensagem é transmitida por um determinado nó, que a retransmite para a estação de destino. No modo de acesso descentralizado, cada nó é responsável por realizar parte do controle. Quanto à confiabilidade, visto que a interface é passiva, a topologia em barra comum oferece maior segurança, pois uma eventual falha em uma interface não afeta o funcionamento da

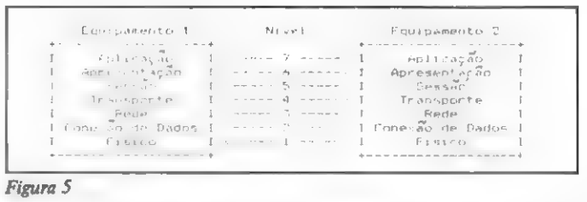


Figura 5

Automação: um caminho para as redes locais

O processamento manual de rotinas administrativas está irremediavelmente condenado na exata proporção em que a Informática vai se tornando mais acessível. A automação do escritório, seja simplesmente para consultas através de terminais não inteligentes, ou mesmo para a execução de tarefas mais simples, como elaboração de folhas de pagamento ou controle de caixa e estoques por intermédio de micros é apenas um passo para a total informatização de procedimentos administrativos.

Micros e minicomputadores, oferecidos em alguns casos a preços atraentes, têm levado empresários a tentarem a experiência de automação de suas firmas. A expansão do número de máquinas deverá ser mais rápida a partir do instante que a concorrência aumente, possibilitando a escolha dos equipamentos em maior variedade e preços mais vantajosos.

Mas o processo de automação deve obedecer a etapas, diz quem já viveu a experiência e que hoje se encontra na fase da rede local, como é o caso de Eraldo de Freitas Montenegro, assistente do chefe do Departamento de Treinamento da Embratel. Para as pequenas e médias empresas a informatização em escala menor é mais eficiente, se analisado o mecanismo de custo/retorno. Rede local deve ser aspiração de quem fez um levantamento pormenorizado de suas necessidades e até de acesso a importantes bases de dados.

O processo de automação de escritórios, seja por intermédio de terminais de consulta ou processamento através de um micro, apresenta características de aspecto psicológico junto ao quadro de funcionários, como pôde observar Eraldo Montenegro no início da implantação da Informática em seu departamento. Mediante um criterioso trabalho, ele passou a observar o comportamento do pessoal da seção 'onde seria implantado o sistema, para conhecer suas reações em função do novo sistema operacional. E constatou fatos que no mínimo, são curiosos.

Por exemplo, havia em parte dos funcionários o temor do desemprego proporcionado pela informatização. Um receio gerado, como ficou comprocado, apenas por questões como ouvi dizer e pela falta de conhecimento pelo menos superficial do significado real da automação.

Descobriu-se então que essa reação era fruto da visão primária a respeito do processamento eletrônico, de que as máquinas seriam ainda aquelas de grande porte, os chamados cérebros eletrônicos, que por sua dimensão transmitiam a falsa imagem do complexo, algo que só pudesse ser acessado por iniciados.

Foi mostrado então a eles que a microeletrônica já possibilitava a fabricação de máquinas de pequeno porte, se não humanas, pelo menos valorizando mais a relação usuário/computador.

A automação de um escritório não deve ser vista apenas como modernidade. *Mas significando dinamização dos trabalhos, eliminação dos feudos e ganhos em termos de produtividade. Isso sem levar em consideração, em tarefas mais rotineiras, aspectos importantes do tipo limpeza, correção e uniformidade, como na correspondência, por exemplo. Esses fatores, no processamento manual, às vezes são confundidos com capacidade profissional, o que não deixa de ser uma avaliação subjetiva.*

Também foi levado em consideração, no exemplo específico da Embratel, que a movimentação de pessoal durante férias ou licença acarreta sempre problemas de atraso devido à necessidade de transferência de atribuições e aprendizado do serviço, o que fica eliminado no escritório automatizado. Em resumo, o domínio das informações não confidenciais é retirado das mãos de uns para ficar à disposição de todos, quando preciso.

Vale então ressaltar que a postura da empresa na hora de optar pela automação deve ser analisada após a pesagem de todos esses aspectos. E a partir do porte de cada uma poderá ser escolhida a simples implantação de micro para processamento interno e com terminais para consultas, até a utilização do sistema rede local, menos simples, porém mais abrangente.

REDES LOCAIS

Para uma empresa que já vive a fase do escritório automatizado, através de elevado número de máquinas espalhadas por seus departamentos, e cujo funcionamento requer constante intercâmbio de informações entre um e outro, justifica-se a implantação da rede local. As rotinas de trabalho, como passagem de memorandos de uma sessão a outra, comunicados internos e alterações de rotinas se desenvolvem de forma mais harmoniosa, eliminando-se a utilização de papéis, e quando necessário seu emprego, isso pode ser feito por meio de impressoras.

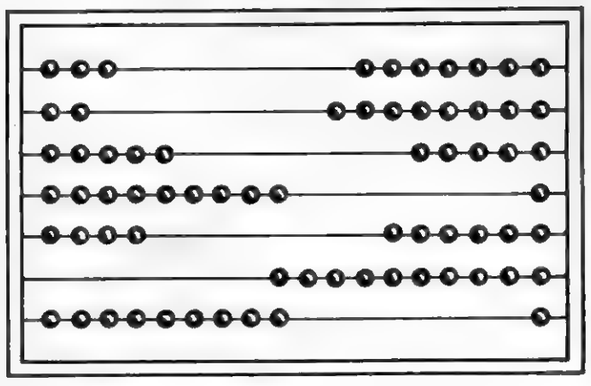
As redes locais permitem o compartilhamento econômico de recursos dispendiosos como unidades periféricas e comportas para bancos e bases de dados externos, dividindo da mesma forma informações que ficam armazenadas após consultas.

No caso específico da Embratel, a implantação da rede local foi feita com a utilização de equipamento adquirido da Cetus Informática, gerando uma rede para operar em seis departamentos. Sua configuração básica é a seguinte: oito postos de serviço, um drive e uma impressora compartilhada; cada posto tem um micro e um nodo CS-1000, servindo de interface entre a linha e a máquina. Dos oito postos, um é operado por um Cobra-305 e os demais por CP-500.

O drive consta de um nodo CS-1200 e dois discos Winchester de 10 Mb cada, para a memória de massa do sistema. O nodo liga a uma Elgin MT-140 serial funcionando em spooling. A linha constitui-se de um par telefônico trançado que interliga postos e servidor totalizando um comprimento de 233 metros.

A rede local da Embratel rodou de início o Correio Eletrônico, desenvolvido em BASIC pelo seu Departamento de Processamento de Dados, possibilitando a troca de mensagens entre os usuários, utilizando um arquivo central localizado nos discos. Mas vai fornecer condições para automação de processos mais complexos ora em desenvolvimento.

É INCRÍVEL O QUE UM BOM PROGRAMA PODE FAZER.



O ábaco, para quem domina sua técnica, permite a execução de contas aritméticas com incrível velocidade.
Da mesma forma, quem possui um microcomputador e um bom programa economiza tempo, papel e aborrecimento.
A Nasajon Sistemas, tem à sua disposição mais de 50 programas como folha de pagamento, crediário, mala direta etc. ... para aproveitar ao máximo o que o seu microcomputador pode oferecer.
Além disso, a Nasajon Sistemas pode desenvolver programas específicos para a sua necessidade, seja ela qual for.
Todos os nossos programas são garantidos e atualizados.
Entre em contato com a Nasajon Sistemas. Estamos sempre dispostos a conversar e esclarecer qualquer dúvida que você possa ter sobre informática.
E quando seu microcomputador estiver funcionando com um programa da Nasajon, você verá as coisas incríveis que ele pode fazer.

nasajon sistemas

Av. Rio Branco, 45 - s/1311 - RJ
CEP: 20090
Tels.: (021) 263-1241 e 233-0615

Você encontra os programas NASAJON também nos seguintes endereços:
Rio de Janeiro: Casa Garson: 252-9191; 325-6458; 541-2345 e 252-2050 - R. 179 – Eldorado Computadores: 227-0791 – Bits e Bytes: 322-1960
Salvador: Officina: 248-6666 - r. 268 São Paulo: Microprocess: 64-0468 - Jundiaí SP – Apoio Com. Informática Ltda: 51-3778 – Tatuí – SP

rede. Nesta topologia, o crescimento também é ilimitado, podendo suportar até 255 estações.

SISTEMAS DE ACESSO

Para que as estações possam trocar dados entre si é preciso um método de acesso que controle a disciplina obedecida pelas estações para acessar o meio de transmissão. Cada método está diretamente associado a um determinado tipo de topologia. Vejamos os mais conhecidos:

No método denominado **Passagem de Permissão** existe uma mensagem de controle, token ou permissão, que é passada de elemento para elemento da rede. Apenas aquele que possui o token pode fazer uso da via de interconexão. Os outros elementos permanecem passivos aguardando a sua vez. A existência de mensagens para controle de acesso nos levam a considerar os seguintes aspectos:

- Overhead da linha, já que a mensagem de controle não transporta informações úteis ■ de processamento, visto que cada elemento da rede deve receber, tratrar ■ passar adiante o token.
- Confiabilidade, pois um erro no meio de transmissão pode tornar a mensagem irreconhecível, ■ se não houver mecanismos que a restaurem, a rede permanecerá inativa até que ela se torne inteligível.

Este método de acesso é normalmente utilizado em redes com topologia em anel.

O método conhecido por **Escaninhos** ou **Slots**, também utilizado em sistemas de topologia em anel, se resume em dividir o anel em escaninhos, que circulam através da rede. Eles são de tamanho fixo ■ possuem um bit que indica se ele está ocupado ou vazio. Para transmitir uma mensagem, a interface aguarda um escaninho vazio, a introduz e seta o bit para indicar que ele está ocupado. Como os escaninhos são de tamanho fixo, a interface deve criar pacotes antes de entrar com os dados na via de transmissão. O controle da rede é centralizado. Existe uma estação responsável pela geração dos sinais necessários. De um modo geral, os mesmos problemas do método de acesso *token passing* estão aqui presentes.

No método **Acesso Múltiplo com Detecção de Portadora – CSMA**, a estação que deseja transmitir verifica antes se existe alguma mensagem fluindo pela via de interconexão. Se houver, aguarda até que a via fique liberada e então envia sua mensagem. Se ocorrer um estado de colisão, ou seja, duas estações enviarem suas mensagens ao mesmo tempo, elas serão superpostas e perdidas. O fato de cada estação verificar se o meio está livre antes de transmitir uma mensagem, já reduz consideravelmente a possibilidade de colisão, já que o tempo de propagação é bem menor que o de transmissão. Entretanto, o tempo perdido com colisões pode ser reduzido com a utilização do mecanismo de detecção de colisão CD. No método CSMA/CD o meio é monitorado antes ■ durante a transmissão de uma mensagem. Neste caso, quando ocorrer um estado de colisão, a transmissão é imediatamente interrompida ■ uma nova tentativa é realizada após um certo intervalo de tempo. Este método é normalmente utilizado em redes de topologia tipo Barra Comum, ■ os problemas citados nos métodos anteriores são aqui praticamente eliminados.

EM SETE CAMADAS

Para redes de computadores geograficamente distantes há um modelo de referência criado pela International Standard Organization (ISO), que consiste em dividir um projeto em sete camadas, relativamente independentes umas das outras (Figura 5).

A denominação do modelo é Open Systems Interconections (OSI) e a descrição de cada nível é ■ seguinte:

- Físico – responsável pela transmissão pura de bits por uma linha de transmissão (voltagens, velocidades, tipo de transmissão etc.).
- Conexão de dados – responsável pelo método de acesso, detecção de erros (protocolo) ■ controle de fluxo.
- Rede – responsável pelo empacotamento de mensagens, ou seja, é transparente ao usuário o tamanho do arquivo a ser enviado.
- Transporte – responsável pela transferência de dados entre equipamentos e pela multiplexação de canais, tornando possível que várias conversões simultâneas ocorram na rede.
- Sessão – Oferece ao usuário ■ acesso à rede, permitindo que dois usuários estabeleçam uma conexão. O estabelecimento de uma sessão envolve a troca de parâmetros.
- Apresentação – responsável pela conversão de códigos, tais como de formatos de arquivos, compressão de texto etc.
- Aplicação – são os programas aplicativos.

Para as redes locais não se formou um padrão devido às particularidades de cada sistema, mas apenas uma recomendação "IEEE-802" que envolve basicamente os níveis 1 e 2 do ISO. É recomendado para meio de comunicação ■ par trançado, cabo coaxial ou fibra ótica. E para método de acesso/topologia as indicações são CSMA/BUS, TOKEN/BUS ou TOKEN/ANEL.

A seleção de uma rede local deve levar em consideração os aspectos já citados e também os seguintes:

- Se é uma rede aberta (aceita vários tipos de equipamentos) ou fechada (requer equipamentos de um só fabricante).
- Características do servidor de arquivos (verificar se possui facilidades para a criação de subdiretórios, controle de acesso por passwords, lock de registros etc.).
- Verificar como a rede local se comporta quando um arquivo já se encontra aberto ■ outra estação executa o mesmo procedimento. Avaliar se o comportamento do sistema, neste aspecto, atende as características particulares de suas aplicações.
- Servidor de impressão (verificar se possui facilidades para determinar prioridades de impressão, se ocorrem superposições de arquivos etc.).
- Verificar se o usuário poderá associar um dispositivo físico (qualquer periférico ligado ao sistema) de uma determinada estação a um dos dispositivos lógicos de sua estação de trabalho.
- Degradação (verificar qual o nível de degradação que ocorre com o incremento de novas estações).

E como conselho final, procure simular todas as situações que deverão ocorrer no momento em que a rede local estiver em operação, para que você não conclua no futuro que não implantou uma rede local, e sim uma rede de problemas.

Amaury Moraes Junior é formado pelo curso de Análise de Sistemas da FASP, tendo feito diversos cursos de aperfeiçoamento nas áreas de Eletrônica Digital e Microprocessadores. Atualmente trabalha na área de microcomputadores para ■ Citybank.

bits

Nova empresa no grupo Prológica

O grupo Prológica está formando uma nova empresa, a CP — Computadores Pessoais LTDA, responsável pela fabricação e comercialização dos computadores pessoais do grupo.

A CP manterá basicamente a infra-estrutura da divisão que existe atualmente, além da ampliação dos departamentos e de maior autonomia e flexibilidade no atendimento a clientes e fornecedores. A totalidade do capital no atendimento a clientes e fornecedores. A totalidade do capital acionário da nova empresa pertencerá aos atuais acionistas do grupo Prológica.

A sede da CP — Computadores Pessoais LTDA ficará na Rua Ptolomeu, 650 — Vila Socorro, São Paulo, CEP: 04762, tel.: (011) 247-6934.

Bolsas de estudo para curso de jogos

A Ciberne Software está oferecendo bolsas de estudo, em regime integral, para programadores interessados no 1º Curso de Projeto e Desenvolvimento de Jogos para Microcomputadores. Os dez bolsistas, que deverão ter mais de 16 anos e serem programadores de equipamentos com processador Z-80, serão selecionados, por entrevista, entre o total de inscritos. O curso, que terá a duração de 50 horas, será ministrado por Renato Degiovani.

Para maiores informações sobre o curso e o procedimento de inscrição, a Ciberne deixa à disposição dos interessados o seu telefone: (021) 262-6968.

Cartões Microcraft

A Microcraft começou o ano com três novos lançamentos para seu microcomputador Craft II Plus. São eles: Cartão Pal/M, Cartão controlador de disquetes de 8" e um drive para discos de 8". Com a nova placa Pal/M o Craft II Plus pode trabalhar com monitor de vídeo ou televisor comum colorido. A placa não vem incorporada no modelo básico do micro, sendo vendida como expansão e seu preço é de 375 mil. Com o cartão controlador de discos, o micro passa a aceitar disquetes de 8", dupla face e dupla densidade, até um total de 4 Mb. Cada placa aceita dois drives de 1 Mb cada e custa Cr$ 1.390 mil. E o novo drive para disquetes de 8", dupla face e dupla densidade, com fonte, cabo de ligação e o próprio gabinete também já está sendo comercializado e seu preço é de Cr$ 8.765 mil. A Microcraft está produzindo atualmente cerca de 200 unidades do Craft II Plus por mês.

Novos jogos Ciberne

A JVA Microcomputadores lançou mais quatro fitas de jogos sob a sigla Ciberne Software. As fitas são dedicadas a equipamentos com lógica Sinclair e, cada uma, contém três jogos que mesclam ação e emoção, criando uma atmosfera de sonho, onde o usuário se transfigura sucessivamente em piloto espacial, mercador, robô e até num cidadão comum à mercê de assaltantes.

Nesse novo grupo, ao contrário do lançado no ano passado, a JVA procurou misturar diversos gêneros de jogos, em cada fita. A intenção foi clara: agradar a todos os tipos de público.

A maioria dos jogos são traduções e versões de jogos americanos, mas a JVA teve a preocupação de manter em cada fita, pelo menos um jogo de autor nacional. Segundo José Eduardo Neves, diretor da empresa, essa iniciativa deverá se tornar uma prática da marca Ciberne.

"Estamos fazendo uma seleção de jogos de nossos autores, com o objetivo de incentivá-los a produzir jogos nacionais. Não nos interessa apenas traduções e versões, mas sim material original. Estamos até promovendo um curso de Programação de Jogos para incentivar o pessoal".

Segundo José Eduardo é possível vislumbrar um maior interesse nessa produção e coisas de qualidade já estão começando a surgir no mercado. Na sua experiência de selecionar esse produto já deu para perceber que a qualidade dos jogos vem crescendo muito. "Tivemos até, há algum tempo atrás, o cúmulo de receber um jogo de autor nacional, totalmente traduzido para o inglês. Segundo o autor isso dava status ao produto!".

Os novos jogos da JVA custam, em média, 2.036 ORTN e, numa primeira fita, um jogo

Valkiria, Mercador dos Sete Mares *e* **Defensor 3D** *são três, das quatro novas fitas da Ciberne.*

nacional de Divino C. R. Leitão dá nome ao produto. **Valkiria** é um jogo de estratégia, do tipo *invaders,* onde o comandante de uma nave espacial tenta, em pleno planeta Vênus, combater estranhas criaturas aladas. Acompanham esse jogo, na mesma fita, a **Guerrilha Cósmica** e o **ZOR**. No primeiro, seres maquiavélicos retiram tijolinhos e vão sendo abatidos, um a um, por um canhão de fótons. É um jogo atraente, do tipo *invaders,* com opção para alta resolução gráfica. Já o **ZOR** é um jogo de ação, que reúne tática a um pouco de sorte. Nele, dois robôs se defrontam no solo de um planeta deserto. Sem estabelecer contato visual, eles se enfrentam com armas e defesas iguais.

Uma segunda fita traz o **Mercador dos Sete Mares** como jogo principal. Este também é um jogo de estratégia, mas não militar, e sim do tipo *banco imobiliário.* No século XIX, o jogador percorre o mundo a bordo de um navio, em busca de ótimos negócios. O seguinte é **Corrida Maluca**, um jogo de ação, tipo PAC-MAN. São dois carros que percorrem um circuito: um tentando apanhar todas as pedrinhas do caminho; o outro, no encalço do primeiro. O último da fita é o **Pinball**, de Divino C.R. Leitão, um jogo que simula na tela uma máquina real de fliperama.

Na fita denominada **Subespaço** está esse jogo, como abertura, simulando uma verdadeira caçada espacial. O jogo é totalmente gráfico e nele o jogador tem que caçar os inimigos que tentam destruir a sua espaçonave. **Cavernas de Marte**, de Divino C. R. Leitão, está a seguir, como um jogo de ação, com um bonito display e cavernas cheias de perigos a serem enfrentados. Por último, nessa fita, está o **Comboio Espacial**, também um jogo de ação, onde uma nave é designada para defender um indefeso cargueiro.

Defensor 3D é a última fita, com naves espaciais que cruzam o espaço em alta velocidade, na mira telescópica de um canhão laser. O próximo é **Q'BERT**, de Divino C. R. Leitão, um jogo que cria um neologismo e utiliza formas geométricas, empilhadas umas sobre as outras, para formar uma pirâmide em perspectiva. O último é **Assalto**, um jogo do tipo PAC-MAN, onde ladrões tentam assaltar um depositante que precisa chegar à salvo no banco.

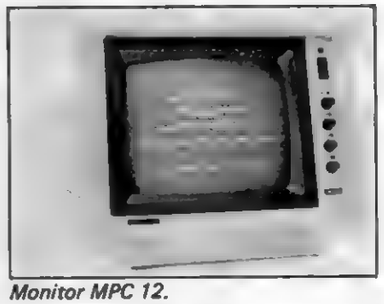
Monitores Videocompo

A Compo está lançando quatro novos monitores de vídeo profissionais: três monocromáticos, que podem ser ligados a computadores que tenham saída de vídeo composto; e um colorido, que traz como novidade uma placa que permite acoplar uma Apple num monitor de vídeo de boa qualidade (a placa converte o sinal do micro em RGB).

O modelo CPC 14 cromático está sendo lançado em 2 versões: média resolução gráfica — 380 x 240 pontos; e alta resolução — 560 x 240. As duas versões são compatíveis com as linhas Apple, IBM e Itautec.

O MPC Vídeo Monocromático é apresentado em 12 e 14 polegadas, sendo que o de 14 é o primeiro deste tamanho a ser lançado no Brasil. As duas opções trazem como novidade a compatibilidade com a placa monochrome IBM, e são compatíveis com as linhas Apple e Itautec. Possuem foco dinâmico, 160 colunas de texto e resolução gráfica de 720 x 240 pontos.

Outro modelo novo, o MV, é apresentado nas versões 1 e 2, ambos monocromáticos. As duas versões apresentam 160 colunas de texto, alta resolução gráfica — 560 x 240 e tela anti-ofuscante opcional. O que as diferencia é que o MV 1 é compatível com a linha Apple e o MV 2 com as linhas IBM e Itautec.



Monitor MPC 12.

O modelo ME Vídeo Monocromático é apresentado em três opções: cinco polegadas e resolução gráfica de 480 x 240; nove polegadas e resolução de 560 x 240, e doze polegadas com reslução de 720 x 240 pontos. As três versões são compatíveis com as linhas Apple, IBM e Itautec.

A Compo oferece seus terminais diretamente ao público e através de revendedores. Informações pelo tel.: (011) 548-6844, São Paulo.

Placa CP/M500, da Microsol

A placa CP/M500, da Microsol, — que possibilita ao CP-500 processar programas no sistema operacional CP/M — está custando menos. A unidade, que custava cerca de 42 ORTN, está agora em torno de 34. Segundo a empresa, isso se deve ao aumento na venda das placas, o que incrementou a produção e, conseqüentemente, barateou o custo da unidade.

A Microsol fica na Av. Pontes Vieira, 1867 — CEP: 60.000, Fortaleza — Ceará.

Relação de Software para TK

A Microdigital está oferecendo uma relação descritiva completa de programas com a marca Microsoft, já desenvolvidos para a linha TK (utilitários, aplicativos profissionais e jogos animados), para que o usuário possa atualizar-se quanto aos programas disponíveis no mercado.

Os interessados devem escrever para: Microdigital Eletrônica Ltda. — Serviço de Suporte ao Usuário — Caixa Postal 54088, CEP 01296, São Paulo, SP.

CCE a Todo Vapor

A CCE entrou o ano de 85 a caminho da concretização daquilo que a empresa havia adiantado no final do ano passado: o lançamento de três novos micros. Em fevereiro, a CCE colocou no mercado o primeiro irmão do Exato, o MC 1000, que veio para concorrer diretamente com os micros pessoais de baixo preço disponíveis no mercado. O novo equipamento foi lançado com suporte de 50 jogos e já estão sendo colocados no mercado mais 100 programas aplicativos desenvolvidos por software houses credenciadas pela CCE. Também já estão disponíveis a expansão de memória de 64 Kbytes, a placa para o MC 1000 rodar programas em CP/M e a interface para utilização de disquetes de 5 1/4", com 170 Kb cada um, face simples e dupla densidade. Para o Exato a CCE colocou no mercado, nos primeiros meses do ano, um monitor de vídeo de 12", fósforo verde ou âmbar (opcional), e as placas CP/M e 80 colunas.

Mas as grandes novidades anunciadas pela empresa ainda estão por vir. Para o segundo semestre está previsto o lançamento do terceiro membro da família de micros CCE, o MC-1500, uma versão ampliada do MC 1000 com gabinete maior e teclado profissional. As interfaces lançadas para o MC 1000 deverão já vir embutidas nesse novo equipamento. Para a Feira de Informática desse ano a CCE promete o lançamento de um micro de 16 bits compatível com o modelo XT da IBM. E na linha de 8 bits a CCE deverá apresentar também um novo equipamento baseado no microprocessador Z-80 e na tecnologia MSX, desenvolvida por um pool de grandes empresas japonesas. O MC 2000 terá memória ROM de 32 Kbytes com uma série de rotinas que facilitarão o trabalho do usuário.

STRINGS

● **A PTI** — Publicações Técnicas Internacionais está promovendo no Brasil o Computer Book Review, periódico americano especializado na análise e crítica de novas publicações na área de processamento de dados. Informações pelo tel.: (011) 258-8442 e 257-1640. ● **A BARTÔ** Computadores Ltda., especialista na área de Commodore, está confeccionando um circuito de proteção contra picos de voltagem que evita queima de equipamentos. Outra novidade da Bartô é a interface RS232 para acoplamento dos computadores da linha Commodore ao projeto Cirandão e outros CBBS. Informações pelo tel.: (021) 262-1213, Rio de Janeiro. ● **A Eastman Kodak Company** anunciou planos para atuar no mercado de telecomunicações, criando uma nova divisão, a Eastman Comunications, para comercializar serviços de telecomunicações. Os serviços iniciais incluirão transmissão de dados e telefonemas a longa distância e, ainda, serviços em "network". ● **A PROLÓGICA** ganhou uma concorrência para fornecimento de microcomputadores ao Ministério do Exército. Para a fase inicial do projeto o Ministério do Exército já recebeu da Prológica 61 Super Sistemas 700 e 74 impressoras P-720. ● **A COMPUSHOP** está aceitando micros usados como parte de pagamento na aquisição de um novo sistema. Os equipamentos serão avaliados de acordo com o estado de conservação e marca por profissionais especializados da empresa. Outra novidade é a comercialização de equipamentos usados, com garantia de três meses. ● **A 3I INFORMÁTICA** já tem vários seminários programados para este ano, entre eles: "Redes Locais ou PBX", "Planejamento Estratégico em Automação de Escritórios" e o "Caminho da Implantação em Automação de Escritórios". Informações pelo tel.: (011) 521-9509, São Paulo. ● **A Texas Instruments** está lançando uma calculadora de mesa que dispensa o uso de baterias. A calculadora — TI-5022 — possui células de captação de energia natural ou artificial, ficando, assim, constantemente ligada. ● **A IBM** está encerrando a medida cautelar de vistoria, que havia apresentado contra a Softec, na Justiça de São Paulo. Isso porque a empresa se comprometeu a não incluir na memória dos equipamentos que fabricar e vender o bios da IBM ou qualquer outro programa a ele semelhante. ● **A Hewlett — Packard Co.** pagou cerca de 65 milhões de dólares em participação nos lucros a mais de 73 mil funcionários, em 32 países. Destes, aproximadamente 240 trabalham na Hewlett — Packard do Brasil Indústria e Comércio, com fábrica em Campinas, São Paulo. ● **A Procada**, empresa de processamento de dados associada ao Grupo Santista, assinou contrato com a Datalógica para distribuição, a nível nacional, dos programas comercializados por esta empresa (dBase II e Framework). ● Um minicomputador **COBRA 530** e um micro **COBRA 210** estiveram presentes na sala de desenho industrial da exposição "Tradição e Ruptura", que se realizou no Pavilhão da Bienal (Parque do Ibirapuera, SP). ● **A Novadata** informa que está desenvolvendo o projeto de seu super-minicomputador, o ND286, que será compatível com o produto atual da empresa, o Mini ND86.

Com a importância que vem assumindo a comunicação entre máquinas o modem ganha papel de destaque como peça fundamental nessa engrenagem

Modems, um periférico em voga

Estabelecer relação, ligar, unir, transmitir. Estes são alguns dos sinônimos encontrados em dicionários para o verbo comunicar, tão em voga em nossos dias. Na área de Informática, o verbo comunicar vem sendo cada vez mais conjugado e mostras disso tivemos na última Feira Internacional de Informática, realizada em novembro, no Rio de Janeiro, onde um dos pontos altos foi o software de comunicação.

O uso do micro como um equipamento isolado esbarra no limite da interação exclusiva entre a máquina ■ seu usuário. Atualmente porém, é cada vez maior o número de usuários de microcomputadores que buscam uma ampliação dessa relação com a máquina, através de ligação em rede ■ da utilização de serviços de bases de dados.

Para que esta ligação se efetue, são necessários três elementos básicos: RS 232-C, um software de comunicação e o modem.

A maioria dos microcomputadores possui saída para ligação de interface RS 232-C que é um tipo de conexão-padrão para a ligação entre os micros e seus periféricos (inclusive modems, para acoplamento à rede telefônica) entre dois ou mais micros e entre um terminal e um computador de grande porte. Esse padrão define como DTE – Data Terminal Equipment, ou Equipamento Terminal de Dados, o equipamento que gera e processa a informação; ■ Data Communication Equipment, ou Equipamento de Comunicação de Dados, aquele que é empregado como transmissor e receptor de dados, no caso o modem.

O segundo componente presente nas ligações entre equipamentos é o software de comunicação. Este software geralmente é comercializado sob a forma de pacotes de comunicação voltados para cada tipo de ligação que se queira efetuar. Existem os pacotes para comunicação entre micros e mainframes, por exemplo, através dos quais o micro passa a atuar como terminal da máquina de grande porte; e os pacotes para acesso às centrais de bancos de dados dos serviços comerciais.

Alguns fabricantes de microcomputadores estão implementando seus equipamentos, dotando-os internamente da interface RS 232-C, do software para acesso à determinados serviços e de modems internos, dispostos em uma placa. Entre estes estão a Itautec, que já dispõe dessa implementação para os micros da família I-7000, para acesso ao Videotexto, e em breve terá disponível também para o I-7000 PC XT, o micro de 16 bits da empresa; a Spectrum, cujos novos Microengenhos também já podem acessar diretamente o banco de dados central da Telesp; e o projeto Cirandão, da Embratel.

O terceiro elemento é o modem, cuja descrição e considerações técnicas se encontram no artigo A viagem dos dados, pág. 10.

O MERCADO DE MODEMS NO BRASIL

A grande maioria dos fabricantes de modems está otimista com a visível expansão desse mercado e com as perspectivas para este ano que são de um crescente aumento no volume de vendas. "Vivemos no rastro da indústria de Informática, já que a indústria de modems é o elo de ligação entre duas áreas: a Informática e as Telecomunicações, possibilitando que estas interajam", afirma Henrique M. Tanabe, supervisor de vendas da Moddata/Coencisa.

A Coencisa é o fabricante que possui a maior variedade de modems no mercado, hoje com 14 modelos diferentes. Criada em 1975, a empresa foi também o primeiro fabricante de modems no Brasil e em 1983 teve seu controle acionário comprado pela Moddata, que na época também já se dedicava à fabricação desses equipamentos.

No primeiro ano de atuação conjunta Moddata/Coencisa, foram vendidos 35% a mais em quantidade e 40% a mais em volume do que no ano anterior, já descontada a inflação. Em 1984 a empresa vendeu, somente em São Paulo, cinco mil modems com um faturamento de Cr$ 35 bilhões em todo o País, e para este ano as previsões são de no mínimo dobrar as vendas. Segundo Henrique Tanabe, estas previsões se concentram principalmente na área de modems para micros, já que segundo ele os equipamentos de alta velocidade mantêm um crescimento constante.

A Moddata/Coencisa fechou o ano com a assinatura de um contrato com a Embratel da ordem de Cr$ 20 bilhões para o fornecimento de modems. Durante o ano de 1984, a empresa investiu cerca de Cr$ 2 bilhões em pesquisa e desenvolvimento ■ hoje os modems da Moddata/Coencisa possuem um índice de nacionalização de 95%, sendo que muitos já atingiram 100%.

A Elebra, conhecido fabricante da área de Informática, possui sete modelos de modems disponíveis no mercado. Em 1984, o Departamento de Transmissão de Dados passou por uma reestruturação na qual foi dada ênfase ao segmento de modems com a criação de serviços para maior suporte técnico. Antigamente, este segmento da empresa era voltado principalmente para grandes usuários e grandes vendas e hoje volta-se também aos pequenos usuários. Segundo o supervisor de planejamento de produto, Alexandre Reznik, o mercado de modems está em franco desenvolvimento, incentivado por clubes de usuários ■ serviços como o Cirandão e Videotexto. "O mercado é promissor e os resultados têm sido excelentes", diz ele, "o teleprocessamento vai crescer com a indústria e comércio comprando cada vez mais modems impulsionados pelo domínio da máquina por parte do usuário".

O modem de maior venda entre os modelos da Elebra é o Banda-Base, um modem digital para ligações urbanas que está sendo muito utilizado por bancos para ligações de terminais entre as agências. A nacionalização dos modems da Elebra varia entre 85 e 100%, este último índice alcançado nos modelos de baixa velocidade, já que para os de alta velocidade ainda são importados determinados componentes. Durante o ano de 1984 a Elebra registrou um crescimento da ordem de 25 a 30% na área de modems e para esse ano está previsto um crescimento também nesta faixa.

No entanto, na maioria dos casos de ligação de micros entre si ou de micros a equipamentos de grande porte os modems utilizados são do tipo analógico e assíncrono. Entre os modems analógicos assíncronos, os de 300 bps são considerados ideais para ligações com microcomputadores pessoais principalmente pelo preço reduzido e pela facilidade de instalação.

Mais especificamente, os modems de acesso a bancos de dados ainda podem possuir um dispositivo de resposta automática (DRA) que atende as chamadas através de um ruído, informando ao usuário que a ligação foi completada. Alguns modelos trazem um outro dispositivo de auto discagem. O usuário programa seu micro com o número que deseja discar e ele o faz automaticamente através do modem.

A opinião de que os serviços de bancos de dados como Cirandão e Videotexto entre outros vêm impulsionando significativamente o mercado de modems é ressaltado por Adailton Souza de Oliveira, Assistente de Marketing da CMA Indústria Eletrônica. A empresa surgiu da CMA Sistemas, que para transmissão de dados dos Estados Unidos para cá começou a fabricar os equipamentos que necessitava, passando a comercializá-los um ano depois. Hoje, a CMA dispõe de cinco modelos de modems, com um índice médio de nacionalização de 90.8%, sendo o A 217 CT o mais procurado para utilização em acesso ao Cirandão e Videotexto. A empresa fabricava um acoplador acústico, hoje fora de linha por falta de mercado.

Outro fabricante pioneiro na área de modems é a Parks, que há 18 anos começou fabricando alarmes residenciais e comerciais e entre 1975 e 76 entrou na área de Informática passando a fabricar modems. Segundo Jaiter Pereira de Pádua, da área comercial, a Parks é uma das três maiores empresas desse segmento juntamente com a Moddata/Coencisa e a Elebra. Ele afirma que 1984 foi um ano muito bom para a Parks, que mesmo com a crise registrou um crescimento real entre janeiro e junho, quando encerra o ano fiscal, de 79%, ■ um faturamento de Cr$ 3 bilhões. Para 85 a meta da empresa é atingir um crescimento da ordem de 150%.

De julho de 1983 a junho de 84 a Parks fabricou 5.300 modems e para o próximo exercício espera, no mínimo, dobrar essa produção. Foram investidos Cr$ 600 milhões em pesquisa e desenvolvimento e ■ índice de nacionalização dos produtos da Parks atinge hoje 98%, com a importação apenas dos circuitos que não são fabricados no Brasil.

Texto final: Stela Lachtermacher

Tabela de modems

A seguir, publicamos uma tabela com os modems que estão no mercado e que são utilizados em microcomputadores. Estes são os endereços das empresas que fazem parte desta tabela: **Moddata/Coencisa** *– R. Dr. Sodré, 72, SP, tel.: (011) 543-2711;* **Elebra Eletrônica** *– Av. Engº Luiz Carlos Berrini, 1461, SP, tel.: (011) 533-9977;* **Parks Equipamentos Eletrônicos Ltda.** *– Av. Parand, 2335, PA, tel.: (0512) 42-5500;* **Digitel Equipamentos Eletrônicos Ltda.** *– R. João Abott, 503, PA, tel.: (0512) 32-5999;* **CMA Indústria Eletrônica** *– Av. Giovanni Gronchi, 6065, SP, tel.: (011) 548-2249;* **ABC Dados Sistemas S/A** *– Estrada do Tindiba, 1608, RJ, tel.: (021) 392-8585;* **CMW Sistemas Ltda.** *– R. José Oliveira Coutinho, 70, SP, tel.: (011) 826-6444;* **Splice Indústria e Comércio** *– Av. Juscelino K. de Oliveira, 154, Votorantin, SP, tel.: (0152) 43-1316;* **Milmar Indústria e Comércio Ltda.** *– Av. Dr. Cardoso de Mello, 1336, SP, tel.: (011) 531-3433;* **EES** *– Rua Napoleão de Barros, 593. SP, tel.: (011) 571-0782.*

ORT
INSTITUTO DE TECNOLOGIA ORT
CENTRO DE INFORMÁTICA

PROGRAMAS DE TREINAMENTO

ÁREA DE MICROINFORMÁTICA

- PARA USUÁRIOS (INTRODUÇÃO A PD, VISICALC/SUPERCALC, WORDSTAR, dBASE II, BASIC)
- PARA PROGRAMADORES E ANALISTAS (INTRODUÇÃO AO MICRO, CP/M, BASIC SOB CP/M, WORDSTAR E dBASE II)
- NA EDUCAÇÃO (LOGO PARA EDUCADORES E PSICÓLOGOS; LOGO PARA JOVENS)

FORMAÇÃO PROFISSIONAL EM PROGRAMAÇÃO E ANÁLISE DE SISTEMAS

APERFEIÇOAMENTO PROFISSIONAL EM AMBIENTE IBM

CPD-ORT: IBM 4341 COM TERMINAIS LABORATÓRIO DE MICROS

TREINAMENTO IN HOUSE EXCLUSIVO PARA EMPRESAS

SOLICITE INFORMAÇÕES E FOLHETOS EXPLICATIVOS

RUA DONA MARIANA, 213 · BOTAFOGO · RJ · TEL · 286-7842

SÓ PARA EMPRESÁRIOS MUITO INTELIGENTES...

— A sua contabilidade atende ■ você ou somente aos fiscais?
— Você tem um bom controle de contas a pagar e a receber?
— A sua administração de imóveis é realmente eficiente?

Na TESBI Informática você encontra programas de contabilidade CAP/CAR e Administração de Imóveis voltados para você, Gerente eficiente. Todos desenvolvidos em DBII ou Basic.
Cursos práticos de dBase, Wordstar e Supercalc.

Melhores Informações pelo tel.: 284-6949 c/Liege

TESBI INFORMÁTICA LTDA.
Av. 28 de Setembro, 226
Lj. 110—V. Isabel

MODEMS DISPONÍVEIS NO MERCADO

FABRICANTE	MODELO	TIPO	VELOCIDADE (bps)	MODO DE TRANSMISSÃO	PREÇOS (ORTN)	SERVIÇOS/ OBSERVAÇÕES
MODDATA /COENCISA	BBC III	sinc./digt.	9600, 4800, 2400 ou 1200	s. d. a 2 ou 4 fios e d. a 4 fios	105	–
	96 BA	assinc./digt.	até 9600	s. d. a 2 ou 4 fios e d. a 4 fios	44	–
	24 TTL-C	sinc./analg.	2400 ou 1200	s. d. a 2 ou 4 fios e d. a 4 fios	170	–
	MC-16	assinc./analg.	até 1600	d. a 4 fios ou s. d. a 2 fios	65	Ciranda
	MC-22	sinc. ou assinc./ analg.	sinc. em 600 ou 1200; assinc. em até 300, 600 ou 1200	d. a 2 fios	148	resp. aut.
	MC-23	assinc./analg.	600, 1200 ou 1200/75	d. a 4 fios ou s. d. a 2 fios	120	Vdt., Cirandão/resp. aut. opc.
	MPC-12	assinc./analg.	até 1200	s. d. a 2 fios	37	Ciranda
	MC-13	assinc./analg.	1200 ou 1200/75	s. d. a 2 fios	25	Videotexto, Cirandão
	MC-31	assinc./analg.	300, 600, 1200 ou 1200/75	d. ou s. d. a 2 fios	39	Ciranda, Cirandão, Videotexto/*1
	300 TTL	assinc./analg.	até 300	d. ou s. d. a 2 fios	71	Cirandão/resp. aut.opc.
	MPC-03	assinc./analg.	até 300	d. ou s. d. a 2 fios	37	Cirandão
ELEBRA	DS-2401	sinc./analg.	2400	s. d. a 2 ou 4 fios e d. a 4 fios	163	resp. aut. opc.
	DA-1201	assinc./analg.	até 1200	d. a 4 fios ou s. d. a 2 ou 4 fios	86	resp. aut. opc.
	DA-1031	assinc./analg.	até 300	d. ou s. d. a 2 ou 4 fios	93	resp. aut. opc.
PARKS	UP-9.600	sinc./digt.	9600	d. a 4 fios ou s. d. a 2 fios	60	–
	UP-2410/S	sinc./analg.	1200 ou 2400	d. a 4 fios ou s. d. a 2 fios	110	série com opc.
	UP-1210/II	assinc./analg.	até 1200	d. a 4 fios ou s. d. a 2 fios	60	série com opc.
	UP-1200 VTX	assinc./analg.	1200/75	d. a 4 fios ou s. d. a 2 fios	25	Videotexto
	UP-1275 VTX	assinc./analg.	1200 ou 1200/75	d. a 4 fios, s. d. a 2 fios	35	Cirandão, Videotexto, Renpac
	UP-310/II	assinc./analg.	até 300	d. a 4 fios ou a 2 fios	55	Cirandão/série com opc.
	UP-9.600	assinc./digt.	até 9600	d. a 4 fios ou s. d. a 2 fios	–	em lançamento
	UP-1200	assinc./analg.	até 1200	d. a 2 fios	–	em lançamento
DIGITEL	AD 9.600 BC	assinc./digt.	até 9600	d. a 4 fios ou s. d. a 2 fios	–	–
	AD 9.600 B	assinc./digt.	até 9600	d. a 4 fios ou s. d. a 2 fios	–	–
	SD 9.600 BC	sinc./digt.	1200, 2400, 4800, ou 9600	d. a 4 ou s. d. a 2 fios	–	com equalizador aut.
	SA 2400 B	sinc./analg	2400	d. a 4 fios ou s. d. a 2 fios	–	–
	AA 1200	assinc./analg.	600, 1200 ou 1600	d. a 4 fios ou s. d. a 2 fios	–	–
	AA 1200 B	assinc./analg.	até 1200	d. a 4 fios ou s. d. a 2 fios	–	–
	AA 2203	assinc./analg.	até 1200	d. a 2 fios	–	com teste remoto
	AA 1275 BC	assinc./analg.	1200 ou 1200/75	d. a 2 ou 4 fios ou s. d. a 2 fios	38	Videotexto
	AA 0302	assinc./analg.	até 300	d. a 2 fios	–	resp. aut. opc.
CMA	A 217 CT	assinc./analg.	1200/75	d. a 2 fios	22	Videotexto/pino DIN
	A 217 C	assinc./analg.	1200/75	d. a 2 fios	28	Videotexto, Cirandão
	A 271 C	assinc./analg.	75/1200	d. a 2 fios	30	–
	A 212 C	assinc./analg.	75/1200	d. a 2 fios	36	Videotexto, Cirandão
	A 230 C	assinc./analg.	até 300	d. a 2 fios	48	–
ABC DADOS	ABC-24	sinc./analg.	2400 ou 1200	d. a 4 fios ou s. d. a 2 ou 4 fios	110	Trans, Renpac/resp. aut.
	ABC 3/12	assinc./analg.	300, 600, 1200 ou 1200/75	d. a 4 fios e s. d. a 2 ou 4 fios	80	Ciranda, Cirandão, Vdt./resp. aut.
CMW	MDA-1200P	assinc./analg.	até 1200	d. a 4 fios ou s. d. a 2 ou 4 fios	–	–
SPLICE	MA-1200	assinc./analg.	1200 ou 1200/75	d. a 2 fios e s. d. a 2 fios	–	Videotexto, Cirandão
MILMAR	Modelo 1	assinc./digt.	2400, 1200 ou 1200/75	d. ou s. d. a 2 fios	–	Videotexto/resp. e disc. aut.
	Modelo 2	sinc./digt.	2400, 1200 ou 1200/75	d. ou s. d. a 2 fios	–	Videotexto/resp. e disc. aut.
	Modelo 3	assinc./digt.	2400 ou 1200	d. ou s. d. a 2 fios	–	resp. e disc. aut.
	Modelo 4	sinc./digt.	2400 ou 1200	d. ou s. d. a 2 fios	–	resp. e disc. aut.
EES	EES-07	assinc./analg.	até 300	s. d. a 2 fios	9	*2

ABREVIATURAS UTILIZADAS
analg. – analógico **assinc.** – assíncrono **aut.** – automática **d.** – duplex **digt.** – digital **disc.** –discagem **opc.** – opcional **resp.** – resposta **s.d.** – semi duplex **sinc.** – síncrono **Trans.** – Transdata **Vdt.** – Videotexto
*1 Este modelo, além do CCITT, é baseado no padrão Bel.
*2 Este modem é o primeiro modelo nacional específico para a linha Sinclair.

unitron
a base de um sistema
inteligente

Quanto mais complexo for um sistema, mais sólida e confiável deve ser sua base. Quando você tem um micro da Unitron como princípio inteligente, você também tem a certeza de que o atendimento de suas necessidades em processamento de dados está assegurado. É a palavra de quem trabalha continuamente para oferecer uma tecnologia sempre atual ao usuário. É o que os fatos demonstram. Na sua categoria, o Unitron andou sempre na frente. Além de contar com uma infinidade de programas, testados e aprovados, e os mais diversificados acessórios de expansão – entre módulos, interfaces e periféricos –, o Unitron agora pode ser conectado, via telefone, a todas as redes existentes: Aruanda, Cirandão, Interdata, Cyber, Videotexto, CMA, etc. Ou, então, às redes particulares, acessando outros micros ou comunicando-se com computadores de grande porte, na função de terminal inteligente. Portanto, se você deseja um processamento de dados com qualidade, fale com nossos revendedores autorizados. Para cada caso, uma solução inteligente. Do princípio ao fim.

unitron
Computadores
CAIXA POSTAL 14.127 – SÃO PAULO – SP – TELEX (011) 32003 UEIC BR

Mais do que um incrível jogo em Assembler e BASIC para seu micro TRS-80: Quasar IV reúne três programas em um só e ainda dá de presente um compilador BASIC, em BASIC, para os mais exigentes. Entre logo nos quadrantes da Galáxia e comande esta ficção!

Quasar IV: uma aventura compilada

Lávio Pareschi

Um bom jogo é difícil. Em BASIC, então, é raro, principalmente por causa da lentidão do interpretador. E na maioria dos jogos em linguagem de máquina (Z80, neste nosso caso), o que temos? Uma luta de reflexos contra uma máquina que não se cansa de repetir a mesma cena, o mesmo movimento, a mesma seqüência... repetir, repetir e repetir. Passada a novidade, não há mais graça. Descobertos os macetes, não existe mais desafio e o único objetivo passa a ser aumentar o número de pontos, o que, convenhamos, é muito pouco.

A maior fraqueza de um jogo é a falta de criatividade, a não variedade, e a própria limitação dos recursos disponíveis impõe esta condição. Simulações de jogos *inteligentes*, como dama, xadrez ou gamão, são exceções, isto quando (e só quando) são bem feitas, o que não impede, no entanto, que muitos os considerem como jogos molengas, justamente devido a ausência de variedade.

E o que é o Quasar IV? Um jogo molenga? Não. Um jogo de puro raciocínio abstrato ou de movimentação assombrosa? Também não. Ele é um jogo em que a principal característica é a variedade, com vários jogos em um, todos seguindo um tema comum, e que ora requer sorte, ora malandragem, ora rapidez, dependendo das circunstâncias. Enfim, um jogo fácil, difícil e impossível.

Neste programa de ficção a grande aventura é sobreviver até o fim dos tempos como comandante de uma espaçonave (a Enterprise, é claro!) que navega pelos quadrantes da galáxia e precisa evitar ou combater os inimigos da Federação (Klingons, Romulans etc.) que não dão sossego. O Quasar IV é em tempo real (centons) e se o comandante não fizer nada, o tempo corre, os inimigos o cercam e.... já era!

Não é fácil ser comandante desta Enterprise: é preciso aprender a lidar com phasers, mísseis, campos de força, tempestades iônicas, buracos negros, minas hiperespaciais, planetas de anti-matéria, sensores inter-galáticos, quadrantes do espaço sideral, comunicações condificadas, chuvas de partículas, módulos de sobrevivência, deformações repentinas do espaço e muito mais. O comandante que chegar (vivo) ao fim de sua missão, no tempo estipulado, receberá uma condecoração especial da Federação!

O jogo é sonoro, com músicas, tiros e ruídos de toda espécie (o que não falta é barulho). Apresenta também efeitos visuais e imagens diversas, representando cada circunstância, e os comandos possíveis são geralmente apresentados na própria tela, como opções para o comandante.

ESTRUTURA DO QUASAR IV

O comando desta espaçonave exige cerca de 24 Kb de BASIC misturados com diversas sub-rotinas Z80, sendo duas compiladas, de aproximadamente 5 Kb cada uma. Fácil, não?

O Quasar IV está dividido em três módulos: 1 – Quasar/MIX, em BASIC (listagem 4), programa principal que carrega os outros módulos e contém todas as sub-rotinas Z80, controlando as chamadas via USR; 2 – QuasI/USR, em Z80, compilado do programa QuasI/CMP, em BASIC (listagem 2), pelo compilador Compiler/BAS (listagem 1), que também será apresentado como parte integrante deste artigo; 3 – QuasII/USR, em Z80, compilado do programa QuasII/CMP, em BASIC (listagem 3), pelo compilador Compiler/BAS.

A configuração mais apropriada para o desenvolvimento do Quasar IV é um equipamento compatível com o TRS-80 modelo III, com 48 Kb de memória e, pelo menos, um drive. As listagens apresentadas do Quasar/MIX e do Compiler/BAS são para o NEWDOS 2.2 (ao final do texto, no Apêndice B, estão as adaptações necessárias para que rodem em TRS DOS).

Para a inclusão das sub-rotinas Z-80 dentro do BASIC de Quasar/MIX, aconselhamos o uso do programa utilitário Pokodes/BAS (publicado em MS nº 36), pois facilita bastante. Por falar em facilitar, com esta intenção o Quasar IV foi dividido em quatro partes: 1º – Compiler/BAS, com descrição, instruções detalhadas e listagem; 2º – QuasI/USR, com listagem (/CMP) e compilação; 3º – QuasII/USR, com listagem (/CMP) e compilação; 4º – Quasar/MIX, sua descrição, inclusão das sub-rotinas Z80, instruções do jogo e listagem.

COMPILER/BAS, O COMPILADOR BASIC, EM BASIC

Uma ferramenta poderosa, o compilador. Para se elaborar um programa decente, e decente tem que ser em linguagem de máquina, pelo menos em parte (como é o caso de Quasar IV), sem que fiquemos loucos ou desesperados escrevendo diretamente em Assembler e debugando durante semanas (isso com otimismo), é essencial a utilização de um compilador.

Mas como conseguir um bom compilador sem gastar muito ou se arriscar com piratas sem manual? Difícil. Porém Quasar IV precisava de um e o jeito foi fazer um, simples, razoável e que não gastasse *kilos* de bytes para somar três números, como pude constatar em certos compiladores que experimentei – uma piada o desperdício de memória!

Compiler/BAS produz código de máquina usando as rotinas da ROM do micro. Não desperdiça memória reservando espaço para variáveis que não serão utilizadas, e no menu inicial são determinadas as dimensões de todos os parâmetros necessários ao programa. Embora isto implique em maior lentidão na execução, ganha-se em espaço de RAM. Rotinas envolvendo números inteiros e gráficos (**SET, RESET, POINT...**) são aceleradas de 50 a 100 vezes e rotinas com números de simples precisão são aceleradas de três a 20 vezes em relação ao interpretador. Por ser escrito em BASIC, sua operação é lenta, gastando um minuto para compilar um Kb de programa, mas o resultado vale a pena. (Quem quiser compilar o próprio compilador, pode tentar...)

Ele trabalha com números e variáveis inteiros ou de simples precisão, até duas dimensões, bem como strings, e compila também os principais comandos do interpretador: **LET, PRINT, IF... THEN... ELSE, GOTO, GOSUB, RETURN, FOR... NEXT, INPUT, POKE, PEEK, SET, RESET, POINT, CLS, REM, END, DEFUSR, USR, OUT, CHR$, VAL** e as funções **RND (0), SQR, ABS, LOG, EXP, COS, SIN, TAN, ATN, INT** e **COS**. Claro que há limitações: é preciso atender a certas exigências na forma de escrever o BASIC para ser compilado e a manipulação de strings poderia ser ampliada (à vontade...), mas o compromisso trabalho versus benefício já atingiu a um bom ponto com o que este compilador é capaz de realizar. Pode-se também trocar variáveis entre um programa em BASIC rodando e um programa compilado chamado por aquele via **USR**, tornando o compilador muito útil em programas mistos, onde rotinas gráficas (jogos) ou recursivas (matemática) são muito lentas quando executadas pelo interpretador.

Existe ainda a facilidade de se testar e debugar primeiro o programa em BASIC, já pronto para a sintaxe de Compiler, que é um subconjunto da sintaxe do interpretador, e, quando funcionando a contento em BASIC, compilá-lo (isto pode parecer elementar, mas tem muito compilador no mercado que não permite).

A listagem 1 contém o programa. Sua numeração deve estar sempre abaixo de **1000**, pois acima de **1000**, inclusive, deve ficar co-residente o programa que se quer compilar. O Compiler/BAS compila tudo o que estiver entre **1000** e **9999**, sendo aconselhável terminar com **10000 END**. Normalmente se faz o merge dos programas, digita-se **RUN** e o Compiler então procura a primeira linha maior ou igual a **1000** e pede os parâmetros do programa a compilar, para o dimensionamento interno das variáveis do programa e, inclusive, da posição em RAM onde se quer colocar o resultado (DUMP e Entry-Point).

Variáveis e armazenamento

O Compiler divide a área de operação do programa compilado em três setores na RAM, de baixo para cima (**0** a **FFFFH**): variáveis, programa e textos. Como stack é usada a pilha do DOS. Quando se faz um **DUMP** para salvar o resultado, é suficiente guardar o programa e o texto, pois a área de variáveis é preenchida na execução do programa. Não se pode esquecer que a proteção do *memory size* deve estar um byte abaixo do início da área das variáveis, se o programa for chamado pelo BASIC. O Compiler/BAS, em sua finalização, fornece as instruções e parâmetros necessários. Existem quatro tipos de variáveis aceitas e armazenadas em posições fixas na área de variáveis:

– *Integer Variables* (**IV**): 26 variáveis possíveis, de A% a Z%, sendo que cada uma ocupa dois bytes de memória.
– *Single Precision Variables* (**SPV**): são 286 variáveis possíveis, que vão de A a Z, A0 a Z0, A1... Z1,..., até A9 a Z9. Cada variável ocupa quatro bytes sucessivos na memória.
– *String Variables* (**SV**): com 26 variáveis possíveis, de **A$** a **Z$**. Cada uma ocupa os bytes na área de variáveis definidos como comprimento das SVs.
– *Single Precision Arrays* (**SPA**): são possíveis 26 variáveis de uma dimensão (**1–D**), de A(IV) a Z(IV); e 26 variáveis de duas dimensões (**2–D**) – quadradas, de A(IV1, IV2) a Z(IV1, IV2). Cada variável ocupa quatro bytes sucessivos na memória. É importante lembrar que as SPAs de duas dimensões têm que ser quadradas.

O limite da dimensão é a memória do micro. Pode-se usar um *array* além de sua dimensão contanto que não se utilize os *arrays* subseqüentes da área de variáveis, ou seja: se **DIM= 20**, é possível usar **A(IV)** na dimensão 30, desde que não se utilize o **SPA B(x)**, que terá seu espaço de memória ocupado por **A(x)**. Normalmente, ao se dimensionar os SPA de uma dimensão, por exemplo, em 10, serão usados os SPAs cujas IVs sejam de 0 a 9.

O compilador aceita a variável A(I%), pois I% é uma IV. Mas não aceita A(2), e se esta for empregada, surgirá uma mensagem de erro. É interessante observar que as variáveis A, A%, A(IV) e A(IV, IV) são diferentes.

Definições e abreviações

- *Integer Variable* (**IV**): A%... Z%.
- *Single Precision Variable* (**SPV**): A. . .Z, A0. . .Z0, . . ., A9. . .Z9.
- *Single Precision Array* (**SPA**): A(IV)... Z(IV), A(IV, IV)... Z(IV, IV).
- *String Variable* (**SV**): A$. . .Z$.
- *Constante* (**C**): qualquer inteiro ou número decimal.
- *Possível Inteiro* (**PI**): qualquer inteiro na faixa de –32767 a 32767.
- *Byte Integer* (**BI**): inteiro de 0 a 255.
- *String* (**S**): qualquer seqüência de caracteres entre aspas, sendo que as aspas finais podem ser omitidas se a string residir no final de uma linha BASIC.
- *Integer Expression* (**IE**): qualquer seqüência da forma **Y1xY2xY3x**. . ., em que Y1, Y2, Y3 representam um inteiro positivo menor ou igual a **32767** ou então uma IV, e **x** pode ser o sinal de + ou de –. A seqüência pode começar com um sinal de subtração, mas não com um sinal de adição ou com um 0 seguido de um sinal de adição. Parênteses não são permitidos, nem necessários. O compilador também avalia expressões inteiras do mesmo jeito que o interpretador, mas se o resultado não estiver entre –**32767** e +**32767**, o programa não indicará o erro e observe-se que **3* 5+1** não é uma IE.
- *Single Precision Expression* (**SPE**): qualquer expressão em BASIC (que não seja ilegal), com ou sem parênteses, formada de: C, IV, SPV, BI, PI, operadores +, –, *, /, símbolo de POT, e as funções **RND(0)**, **SQR**(SPE), **ABS**(SPE), **LOG**(SPE), **EXP** (SPE), **COS**(SPE), **SIN**(SPE), **TAN**(SPE), **ATN**(SPE). Veja a seguir quatro exemplos que ilustram bem:

```
a) -(1+SQR(1.2*A%+SIN(A(I%,J%)*2.5)))
b) LOG(ABS(Z0*0.123+1)/SIN(COS(TAN(A(K%)+1))))
c) (-1.2+3.4+5)
d) (I%+J*L%)
```

O compilador avalia as SPEs da esquerda para a direita, não importando a sua natureza (sejam *, /, +. . .), mas respeita os níveis de parênteses. Por isso, é preciso tomar bastante cuidado com a forma de escrever as SPEs, devendo-se usar os parênteses

à vontade para obter o resultado correto. Assim foi feito para facilitar a elaboração. Por exemplo, uma linha BASIC, assim: **A+(B+2^C)*(D+E)**, terá que ser reescrita para: **(2^C+B)*(D+E)+A** ou então **A+((B+(2^C))*(D+E))**. Se houver dúvida, é preferível usar parênteses.

Comandos e sintaxes

• **LET**: a palavra **LET** não é necessária, mas pode ser usada nas seguintes circunstâncias:
1) *Integer LET:* X=Y
X: IV
Y: IE, **INT(SPE)**, **PEEK(PI)**, **PEEK(IV)**, **POINT**(z, y), em que z e y são IVs ou BIs.
Exemplo:

```
X%=S%+3+INT(2*RND(0))+PEEK(-1)
```

2) *Single Precision LET:* X=Y
X: SPV
Y: SPE
Exemplo:

```
Q=SQR((A*A)+(B*B))
```

3) *String LET:* X=Y
X: SV
Y: S, SV, **CHR$(z)+CHR$(y)**+. . . em que ■ e y são IVs ou BIs.
Exemplo:

```
A$="Compiler", A$=CHR$(32)
```

Observe-se que **A$+X$**. . . ou "abc"+"def" não podem ser usados, entretanto, através de **PRINT**s é possível a concatenação de strings.

• **PRINT**
1) *PRINT* (line feed e carriage return)
2) *PRINTX; Y; Z;*. . .
X, Y, Z: SPEs, SVs e Ss
3) PRINT @ X, Y; Z;. . .
X: PI de 0 a 1023, e IV
Y, Z: SPEs, SVs e Ss
4) *PRINT @ X, Y*
X: PI de 0 a 1023, e IV
Y: **CHR$(BI)+CHR$(BI)**+. . . (neste caso, o cursor não é modificado)
Repare que não se deve usar vírgula, mas sim ponto e vírgula, na separação entre variáveis dentro de um PRINT.

• **IF. . . THEN. . . ELSE**
1) *IFXusgYTHEN (# linha) ELSE*. . .
X: IV, SPE (mas que não comece com uma IV)
usg: =, >, <, <>, ><, >=, <=, =>, =<
Y: se **X=SPE THEN SPE**; se **X=IV THEN** IV ou PI
linha: número de linha BASIC
Note-se que ELSE pode ser seguido de quaisquer outras instruções, inclusive IF. . . THEN. . . ELSE.
2) *IFXusgYTHEN . . . . . :GOTO (#* linha) *ELSE. . . . .*
XusgY: igual a anterior
linha: número de linha BASIC
Neste caso, depois de **THEN** e **ELSE** podem vir quaisquer outras instruções, inclusive **IF. . .THEN. . . ELSE**.
3) *IFXusgYTHEN* (# linha) ou então *IFXusgYTHEN. . . . .* (quaisquer instruções)
É importante destacar as seguintes características: o limite de **IF**s, um dentro do outro, em uma mesma linha, é de 10; **THEN** pode ser substituído por **GOTO** ou **THEN GOTO** (nos itens 1 e 3); diferentemente do interpretador, é preciso terminar cada **THEN**. . ., quando seguido de instruções, com um **GOTO** antes do **ELSE**. Veja três exemplos elucidativos:

```
IFA%=B%THEN2000ELSEPRINT"pqrt"

IFZ%=>X%THENPRINT"Pelé":GOTO2000ELSEEND

IFA<>2*BTHENA%=INT(A):PRINTB
```

Repare que: se X= IV e Y= SPE que não comece com uma IV, pode-se relacionar **(X%)usgSPE** ou **SPEusg X%**. Outra observação interessante: **IF THEN ELSE** com números inteiros é muito mais rápido.

• **GOTO** (# linha)
linha BASIC
• **GOSUB** (# linha)
• **RETURN**
• **FORX=YTOZ**
X: IV
Y, Z: IV, PI
Observação: Y tem que ser < =Z (atenção que o compilador não indica este erro); **STEP** não é aceito, mas pode-se fazer **STEP**s diferentes de 1 criando-se loops de software ou alterando-se x dentro do FORX. . . NEXTX.

• **NEXTX**
X: IV
Observações: X não pode faltar; não se deve pular fora de um loop **FOR-NEXT** sem correr o risco do programa sob execução falhar. Cada **NEXT** deve estar associado ao seu **FOR** antecedente, mas o compilador não indicará se isto não for obedecido. Múltiplos **FOR–NEXT** são permitidos sem limite. Este exemplo demonstra a sintaxe certa:

```
FORI%=1TO10:FORJ=1TO50:...:NEXTJ%:NEXTI%
```

• **INPUTX** ou **INPUT** "...";**X**
X: IV, SPV, SV
Deve-se ressaltar que os números podem ter até seis dígitos. Se X= IV, é possível introduzir números decimais, que serão truncados. Não esqueça que números inteiros para IVs devem estar contidos entre +/– 32767, caso contrário, um erro fatal ocorrerá.
O POKEX, Y
X: IV, PI
Y: IV, BI
• **SET(X, Y)**; **RESET (X, Y)**; **POINT (X, Y)**
X, Y: IV, BI (sendo que dentro da faixa legal para tais funções)
• **OUTX, Y**
X: BI
Y: BI ou IV
• **DEFUSR=X**
X: IV ou PI
• **USR (X)**
X: IV ou PI
É útil destacar que com os comandos acima **(DEFURSR= X** e **USR(X))** um programa compilado pode chamar outro via **URS** ou também rotinas de som, por exemplo:

```
40 DEFUSR=A%:USR(0)
```

• **CLS**; **REM** ou ' e **END**
É preciso que haja um **END** em cada ponto que se quer retornar ao programa chamador do programa compilado (como um **RETURN**). No caso do chamador ser um programa do BASIC via **USR**, é antes do **END** que as variáveis do programa Z80 podem ser transferidas para o programa em BASIC.

• **X=VAL (Y)**
X: IV
Y: SV
Inteiros negativos retornam zero, e inteiros acima de 32767 retornam como – ( ). Por Exemplo: **I%=VAL (A$)**, em que **A$="60000"** retorna **I%=–5536**.

Transferências de variáveis

Como certas funções do interpretador não podem ser compiladas (disco e cassete I/O, **PRINTUSING**, manipulação de strings. . .), é interessante que ao se elaborar um programa misto – BASIC e Z80– haja um meio fácil de se transferir dados em variáveis entre um programa ■ outro. Para isso, o Compiler/BAS utiliza as seguintes instruções:

• X=0+Y Y(BASIC) para X(Z80)
X: IV ou SPV do programa Z80
Y: IV ou SPV do programa BASIC
• X=1^Y X(Z80) para Y(BASIC)
X: IV ou SPV do programa Z80
Y: IV ou SPV do programa BASIC
(Obs.: ambas IVs ou ambas SPVs)

Eis alguns esclarecimentos necessários: se a variável Y do BASIC ainda não existir quando for realizado um **X=1^Y**, o programa compilado usando as rotinas da ROM do micro a criará, efetuando normalmente a transferência; se, ao debugar o programa fonte em BASIC/CMP, for feito um X=Y, não haverá interferência na operação; as transferências podem ocorrer em qualquer ponto do programa, embora em geral sejam feitas no início ■ no fim (antes do **END**).

No menu inicial, os parâmetros

Ao se rodar o Compiler/BAS, este pede uma série de parâmetros que vão dimensionar as áreas das variáveis, do texto e do programa e estabelecer seus respectivos início e fim na memória. O compilador calcula as áreas reservadas de cima para baixo, a partir do topo da memória (**FFFFH-300** bytes de sistema proibidos). Acompanhe a seguir uma breve descrição da seqüência de parâmetros solicitados:

a) Memória para programa?
Estimativa do tamanho (bytes) do programa compilado (área de programa). Como regra geral, estima-se em 1500 bytes de Z80 para cada 1000 bytes de ocupação (não após a execução) do programa fonte (BASIC/CMP).

b) Número de linhas a compilar?
Estimativa do número de linhas do programa fonte em BASIC/CMP.

c) Números de **GOTO**s mais **GOSUB**s?
Estimativa (superior) do número de **GOTO**s e **GOSUB**s existentes no programa fonte.

d) Offset de memória?
Permite reservar espaço adicional no fim da memória, fazendo com que o topo da RAM para o compilador não seja **FFFFH – 300**, mas **FFFFH-300** menos o valor fornecido neste parâmetro. Se for 0, o topo permanece em **FFFFH-300**. Isto é muito útil quando se quer usar vários programas compilados juntos que, é claro, não poderão ocupar a mesma região; ao se compilar o segundo programa, por exemplo, dá-se um offset equivalente à área efetiva (variáveis+programa+textos) do primeiro programa, o que permite ao compilador reservar uma área para ele, no topo da RAM.

e) Bytes para texto?
Estimativa do número de caracteres a serem usados como texto no programa. Texto para o compilador é toda string entre aspas dentro de um **PRINT**. Por exemplo: **PRINT "Pitrusgh"**.

f) Neste ponto, o compilador vai procurar a primeira linha igual ou superior a linha **# 1000** dentro do BASIC onde estão o Compiler/BAS ■ o programa fonte (de **1000** a **9999**). Esta procura demora cerca de meio minuto, e isso é muito importante. (Quem quiser colocar o programa fonte antes do compilador para pegar logo a primeira linha BASIC para compilar, pode fazê-lo, mas vai ter que se preocupar com a virada dos endereços e ponteiros internos de **32767** para – **32768** das variáveis inteiras usadas nos **POKE**s e **PEEK**s do programa, após os 32 Kb de memória inicial. Usando-se o BASIC Disco, depois do compilador a linha 1000 do programa fonte estará certamente após o endereço **32767**.

g) Entry-Point ok (S/N)?
É mostrado o Entry-Point do programa calculado após as áreas já definidas ■ deve-se responder se está ok ou não. Caso queira-se determinar um outro Entry-Point, é só responder não que o menu pedirá o novo endereço decimal, ■ uma nova localização do programa compilado é calculada em função do novo Entry-Point.

h) Número de SPVs com números?
O número de variáveis inteiras possíveis é fixo em 26, de A% a Z%. O número mínimo de SPVs possíveis também é 26 (de A a Z) se for respondido 0, mas existe a possibilidade de se chegar a 286 combinando-se números com letras. Se a resposta for 1, estarão disponíveis as variáveis de A a Z e de A0 a Z0, ■ assim sucessivamente até 10: de A-Z, A0-Z0. . . A9-Z9. Portanto, ao se escolher as variáveis SPVs do programa fonte, deve-se fazê-lo nesta seqüência para não desperdiçar memória, ■ jamais usar A, X1 e Z9. Cada SPV usa quatro bytes e se for preciso utilizar todas as SPVs haverá o emprego de 1144 bytes da memória.

i) Dimensão dos *arrays* de uma dimensão (1–D)?
Seu único limite é a memória disponível. Como os *arrays* de uma e duas dimensões (1–D/2–D) são todos SPVs, cada unidade do *array* precisa de quatro bytes.

j) Número de SPAs de uma dimensão (1–D)?
1:A( ); 2:A( )-B( ); 3:A( )-B( )-C( ); 26:A( ) a Z( )

k) Dimensão dos SPAs de duas dimensões (2–D)?
São sempre quadrados ■ o limite é a memória.

l) Número de SPAs de duas dimensões (2–D)?
1:A(,); 2:A(,)-B(,); 5:A(,) a E(,); 26:A(,) a Z(,)

m)Número de variáveis strings?
1:**A$** 2:**A$**,**B$** 3:**A$**,**B$**, **C$** . . . 26:**A$**,**B$**, ..., **Z$**

n) Comprimento máximo das SVs?
De 0 a 255.

Agora o compilador dá uma geral nos parâmetros e aguarda um sinal após a revisão do usuário e, finalmente começa. Se for encontrado algum erro na sintaxe apresentada, o compilador pára e mostra o número da linha incorreta. A medida em que o trabalho é realizado, algumas informações são exibidas na tela para permitir o acompanhamento da compilação. Se uma das três áreas (variáveis, programa e texto) se sobrepor à outra, é indicado o erro.

Ao terminar a compilação, é apresentado um mapa de endereços da memória utilizada, com início ■ fim de todas as áreas de IVs, SPVs, SPAs de 1-D, SPAs de 2-D, SVs e códigos de máquina com programa e textos. E, por fim, o compilador acaba, oferecendo duas opções:
1 - **RUN**: executa o programa compilado e já na memória;
2 – **SAVE**: mostra no vídeo o **DUMP** necessário para se salvar o programa da memória, com os parâmetros de INÍCIO, ENTRY-POINT e FIM. Note-se que o comando **DUMP** só pode ser executado manualmente, pois não permite variáveis nos seus parâmetros (tanto em TRSDOS quanto em NEWDOS).

Dicas para não errar

É fundamental seguir as regras já descritas, pois o compilador nem sempre indica que há erro na linha tal e, se houver erro e o programa for executado. . . Adeus! Justamente para evitar isso, leia com bastante atenção estas dicas:

– Os comandos corretamente especificados, exceto em um caso de **PRINT @** e de **USR ( )**, deverão ter os mesmos resultados quando rodados em BASIC ou compilados.
– Cuidado especial deve ser dado às expressões de simples precisão (SPE), não se esquecendo a ordem de execução dos

operadores aritméticos, da esquerda para a direita. O uso de parênteses pode facilitar muito.

– As variáveis devem ser **inicializadas** antes de serem usadas pela primeira vez, pois se não são zeradas, contêm lixo.

– Não são aceitos espaços (blanks) em meio aos comandos. Assim, é errado digitar A% =2, o correto é A%=2.

– Múltiplos comandos em uma mesma linha são normalmente permitidos, desde que sejam separados por dois pontos (:), por exemplo: **PRINT "Name"; : INPUTA$.**

– Para passar o valor de uma SPV para uma IV, utiliza-se a função **INT( )**. Exemplificando: **X% = INT(A)** ou então **X% =INT(A*RND(0))**.

– Muita cautela com os parâmetros: coisas muito esquisitas podem acontecer se eles forem mal dimensionados.

– aproveite a velocidade dos comandos **PRINT @ , SET, RESET** e **POKE** para dar *ânimo* ao visual dos seus programas. Sem esquecer de incluir rotinas para controle da velocidade.

– O compilador não verifica se o resultado para uma IV é um inteiro: 40000 não cabe em X% e, assim, não vai funcionar direito.

– Um **GOTO** não deve ser colocado dentro de um **FOR... NEXT**. Aliás, não inventar na programação é sempre um bom conselho.

– Rotinas de som são sempre melhores se geradas em Assembler e não compiladas. Mas, variando **OUT255,X** pode-se obter um resultado razoável.

– Melhor do que empregar o **INKEY$** é usar **PEEK** (endereço do teclado). Fazendo **PEEK(14400)**, tem-se: 1-**ENTER**, 2-**CLEAR**, 4-**BREAK**, 8-**UP**, 16-**DOWN**, 32-**LEFT**, 64-**RIGHT**, 128-**SPACE**. . . (o manual do equipamento deve ter os demais endereços).

E nada melhor do que testar tudo o que foi afirmado sobre o compiler/BAS com este exemplo de aplicação, que roda umas 25 vezes mais rápido do que em BASIC:

```
1000 INPUT"Name";A$
1010 CLS:A%=0:GOTO1100
1020 X%=INT(128*RND(0)):Y%=INT(48*RND(0)
):RETURN
1100 FORJ%=1TO30:OUT255,3:GOSUB1020:SET(
X%,Y%):OUT255,1:GOSUB1020:RESET(X%,Y%):N
EXTJ%
1110 IFA%=0THENPRINT 506,".";A$;".";:A%=
1:GOTO1120ELSEPRINT@506,".........";:A%
=0
1120 FORJ%=1TO70:OUT255,1:FORK%=J%TO77:N
EXTK%:OUT255,1:NEXTJ%'som e delay
1130 J%=PEEK(14400):IFJ%<>4THEN1100ELSEE
ND
10000 END
```

Os parâmetros para compilação do exemplo acima são: uns 500 bytes para o programa (dá e sobra); 10 linhas; 10 GOs; offset zero; 100 bytes para texto; Entry-Point ok; SPVs com número igual a zero (o programa nem usa SPVs); DIM 1=D, número 1-D, DIM 2-D e número 2-D, tudo zero; número de SVs igual a 1 (usou-se só **A$**); comprimento de SVs igual a 10 e... Pronto.

Vamos agora à listagem 1.

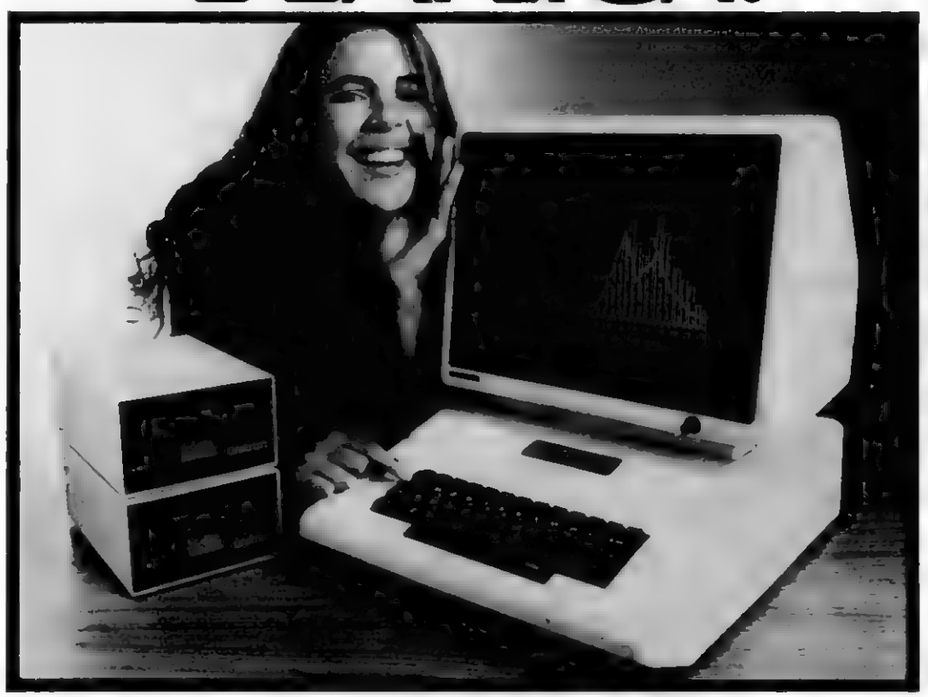
Listagem 1 - Compiler/BAS

```
0 'Basic Compiler. ======= Retire Rems ======= Zorro/84
1 POKE16561,255:POKE16562,255:POKE16544,255:POKE16545,255:CLEAR:
CLS:PRINT@960,"Memoria Disponivel!";MEM-2560;:LI=65536-MEM+2400:
I!=INT(LI/256):POKE16562,I!:POKE16561,LI-256*I! :GOTO149
2 POKEM,P:PRINTP;:M=M+1:RETURN
3 PC=PEEK(Q):PN=PEEK(Q+1):Q=Q+1:IFPC=32THEN3ELSEIFPC=0C=2:RETURN
ELSERETURN 'peek proximos codigos basic
4 PC=PEEK(Q):Q=Q+1:RETURN
5 IFPC(650RPC)90THEN8ELSERETURN
6 IFPN()37THEN8ELSEQ=Q+1:RETURN
7 IFC1)9990RC1(1000OTHENRETURN
8 PRINT:PRINT"ERROR LINE #";L1(L):END
9 PRINT@40,TIME$;:RETURN
10 I=PEEK(FNA(Q+2))+256*PEEK(FNA(Q+3)):RETURN
11 'Rotinas de simples precisao
12 GOSUB3:GOSUB5:V1=PC-65:IFPN(48ANDPN)57ANDPN()213ANDPN()40ANDC
F()1THEN8
13 IFPN)47ANDPN(58MI=PN-47:GOSUB3ELSEMI=0
14 IFPN=213Z1=1:RETURN
15 IFPN=40GOSUB3:GOSUB3:GOSUB5:V2=PC-65:GOSUB6:GOSUB3ELSEIFCF=1Z
1=1:RETURN
16 IFPC()41ANDPC()44THEN8
17 IFPC=41Z1=2:RETURN
18 IFPC=44GOSUB3:GOSUB5:V3=PC-65:IFV1(D0RV1)=NTTHEN8ELSEGOSUB6:Z
1=3:GOSUB3:IFPC()41OR(PN()213ANDCF()1)THEN8ELSERETURN
19 'Calculo Endereco
20 ONZ1GOSUB21,22,23:RETURN
21 IFMI)1STHEN8ELSEC1=VF+(V1+MI*26)*4:GOSUB83:GOTO112
22 V7=V1:V8=V2:GOTO25
23 V7=V1:V8=V2:V9=V3:GOTO26
24 'Arrays
25 V0=V8:GOSUB117:GOSUB114:C1=VA+V7*D0*4:GOSUB83:GOSUB111:GOTO11
3 'i-d
26 V0=V9:GOSUB117:P=41:GOSUB82:C1=VD+4*NT*DT*DT+2*V7*DT:GOSUB83:G
OSUB111:GOSUB113:P=94:GOSUB82:P=35:GOSUB2:P=8&:GOSUB2:V0=V8:GOSUB
117:GOSUB114:GOTO113 '2-d
27 'Avaliacao de Expressoes de Simples Precisao
28 GOSUB3:IFPC()2060TO30
29 E1=0:D1=0:GOSUB112:GOSUB124:GOSUB124:GOTO35 'sinal -
30 GOSUB41
31 GOSUB3:IFC=20RPC=410RPC=580RPC=590RPC=2120RPC=2130RPC=2140RPC
=1410RPC=2D2THENRETURN 'se terminacao, return
32 GOSUB128 'move de (4121H) p/ stack, valor intermediario
33 'poe nova variavel em 4121H, executa
34 IFPC=205GOSUB3:GOSUB41:GOSUB129:GOSUB131:GOTO31 'soma
35 IFPC=206GOSUB3:GOSUB41:GOSUB129:GOSUB132:GOTO31 'subtrai
36 IFPC=207GOSUB3:GOSUB41:GOSUB129:GOSUB133:GOTO31 'multiplica
37 IFPC=208GOSUB3:GOSUB41:GOSUB129:GOSUB134:GOTO31 'divide
38 IFPC=209GOSUB3:GOSUB41:GOSUB129:GOSUB135:GOTO31ELSE8 'exp
40 'Rotina p/ tratar constantes, variaveis e funcoes em expresso
es de simples precisao e coloca-las na area de 4121H
41 IF(PC(58ANDPC)47)ORPC=46GOSUB86:RETURN 'converte constantes p
/ representacao de 4 bytes
42 IFPC=222GOSUB3:IFPC()40THEN8ELSEGOSUB3:IFPC()48THEN8ELSEGOSUB
3:IFPC()41THEN8ELSEP=205:GOSUB82:P=24D:GOSUB82:P=20:GOTO2 'Rnd(D)
43 IF(PC)220ANDPC(229)ORPC=217ORPC=40THEN54ELSEGOSUB5 'funcao ou
 erro
44 V4=PC-65:IFPN)47ANDPN(58THENME=PN-47:GOSUB3:Z2=1:GOTO49ELSEIF
PN()40ANDPN()37THENME=D:Z2=1:GOTO49ELSEIFPN=37THENZ2=4:GOSUB3:GO
TO49
45 GOSUB3:GOSUB3:GOSUB5:V5=PC-65:GOSUB6:GOSUB3
46 IFPC()41ANDPC()44THEN8
47 IFPC=41Z2=2 'i-d array
48 IFPC=44GOSUB3:GOSUB5:V6=PC-65:GOSUB6:Z2=3:GOSUB3:IFPC()41THEN
8 '2-d
49 ONZ2GOTO50,51,52,53
50 IFME)1STHEN8ELSEC1=VF+(V4+ME*26)*4:GOSUB83:GOSUB112:GOTO127
51 V7=V4:V8=V5:GOSUB25:GOTO127
52 V7=V4:V8=V5:V9=V6:GOSUB26:GOTO127
53 V0=V4:GOSUB117:P=34:GOSUB82:P=33:GOSUB82:P=65:GOSUB82:P=205:GOSU
B2:P=204:GOSUB2:P=10:GOTO2 'converte variaveis inteiras em simpl
es precisao
54 IFPC=40THEN64ELSE8=8+1 '(
56 IFPC=221THEN65 'sqr
57 IFPC=217THEN66 'abs
58 IFPC=220THEN67 'log
59 IFPC=224THEN68 'exp
60 IFPC=225THEN69 'cos
61 IFPC=226THEN70 'sen
62 IFPC=227THEN71 'tan
63 IFPC=228THEN72 'atn
64 GOSUB28:GOTO73
65 GOSUB28:GOSUB137:GOTO73
66 GOSUB28:GOSUB138:GOTO73
67 GOSUB28:GOSUB139:GOTO73
68 GOSUB28:GOSUB140:GOTO73
69 GOSUB28:GOSUB141:GOTO73
70 GOSUB28:GOSUB142:GOTO73
```

NÃO PERCA A PRÓXIMA EDIÇÃO DE MICRO SISTEMAS

● CBBSs — como funcionam, o que oferecem, quais os existentes etc. ● Reportagem sobre clubes de usuários e pontos de encontros ● Técnicas de Rede PERT para Apple e Sinclair ● Para TRS-80: continuação do MBDADOS ● No Banco de Software: Controle de congelados, Conta bancária, Funções no CP/M e muito mais.

CLAPPY LANÇA O TI UNITRON E EXPLICA:



T. I. QUER DIZER TECLADO INTELIGENTE.

O TI é o mais novo microcomputador da Unitron.

Ele tem um microprocessador 6504 e um teclado inteligente. Isto é, um teclado gerador de caracteres para a lingua portuguesa. Veja o que este teclado pode fazer:

Um. Programação de funções especiais em qualquer tecla.

Dois. Redefinição das posições da tecla pelo próprio usuário.

Três. Modo de operação igual à máquina de escrever.

Quatro. Repetição automática de caracteres.

Cinco. Diagnóstico de teste automático ao ligar.

Venha conhecer o TI pessoalmente na Clappy ou solicite a visita de um Consultor Técnico Clappy no seu escritório.

Aliás, na Clappy você encontra tudo o que precisa em microcomputadores, periféricos, suprimentos, softwares. Além de cursos próprios de programação e operação, assistência técnica, implantação e instalação de sistemas. E mais.

Aplicativos comerciais: contabilidade, controle de estoque, folha de pagamentos, contas a pagar e a receber.

Aplicativos de apoio: planilha financeira, processamento de dados, mala direta, cadastro e controle financeiro, gráficos, etc.

Seja por venda, seja por leasing, ninguém pode fazer um preço melhor do que a Clappy.

unitron Clappy

Centro: Av. Rio Branco, 12 - loja e sobreloja.
Tel.: (021) 253-3395
Centro: R. Sete de Setembro, 88 - loja Q (galeria)
Tel.: (021) 222-5517/222-5721
Copacabana: Rua Pompeu Loureiro, 99.
Tel.: (021) 257-4398/236-7175
Aberta diariamente das 10 às 20 horas e aos sábados das 9 às 14 horas. Estacionamento próprio.
Assistência Técnica: 234-9929/234-1015
Entregamos em todo Brasil pelo reembolso Varig.

```
71 GOSUB28:GOSUB143:GOTO73
72 GOSUB28:GOSUB144
73 IFPC()4ITHENBELSERETURN
75 'ee Rotinas de Conversao ee
76 'Calculo de Lsb/Msb de string numerico inteiro (Ascii)
77 C$="":IFPC=206THENPC=45:GOSUB81ELSEGOSUB80:IFC$=""THENC1=-1:R
ETURN
78 C1=VAL(C$)
79 D1=C1/256:E1=C1-D1*256:IFC1(OTHEND1=D1+256:C1=-C1:RETURNELSER
ETURN
80 IFPC(480RPC)57THENRETURN
81 C$=C$+CHR$(PC):GOSUB3:GOTO80
82 'Calculo de enderecos acima de 32K
83 D1=C1/256:E1=C1-D1*256:D1=D1+256:RETURN
84 Z=VT+V1+V1:P1=Z/256:P=Z-P1*256:P1=P1+256:RETURN
85 'Conversao de string numerico em simples precisao, 4 bytes
86 C$=CHR$(PC)
87 GOSUB3:IF(PC(58ANDPC)47)ORPC=46C$=C$+CHR$(PC):GOTO87
88 R=VAL(C$):GOSUB89:E1=33:D1=65:GOSUB112:C1=83:GOSUB116:P=35:GO
SUB2:C1=82:GOSUB116:P=35:GOSUB2:C1=61:GOSUB116:P=35:GOSUB2:C1=8E
:GOSUB116:@=@-1:RETURN
89 IFR=0THENBE=0:B1=0:B2=0:B3=0:RETURN
90 Y1=1:Y2=2:N=1:IFY1)RTHEN93
91 IFY2(=RTHENY1=Y1+Y1:Y2=Y2+Y2:N=N+1:GOTO91ELSE94
93 IFR(Y1THENY1=Y1/2:Y2=Y2/2:N=N-1:GOTO93
94 BE=N+128:X1=0:R=R-Y1:GOSUB97:B1=B
95 GOSUB98:X1=X:GOSUB97:B2=B
96 GOSUB98:X1=X:GOSUB97:B3=B:RETURN
97 GOSUB98:X2=X:GOSUB98:X3=X:GOSUB98:X4=X:GOSUB98:X5=X:GOSUB98:X
6=X:GOSUB98:X7=X:GOSUB98:X8=X:B=X1+X1+X2+B+B+X3+B+B+X4+B+B+
+X5+B+B+X6+B+B+X7+B+B+X8:RETURN
98 Y1=Y1/2:R=R-Y1:IFRT(0X=0:RETURN:ELSEX=1:R=RT:RETURN
99 'Poke string na area temporaria
100 C1=MF:GOSUB83:GOSUB112:NN=1
101 IFPC=34GOTO1D9
102 GOSUB108
103 IFFP=1ANDPEEK(@)=32@=@+1:PC=32:NN=NN+1:GOTO103
104 GOSUB3:NN=NN+1
105 IFFP=1AND(PC=34OR PC=2)GOTO1D9
106 IFFP=0AND(PC=36ORC=2)GOTO1D9
107 GOTO102
108 C1=PC:GOSUB116:IFPC()0THENP=35:GOTO2ELSERETURN
109 @=@-1:C=D1:PC=3:GOSUB108:IFNN)8LTHENPRINT:PRINT"STRING TOO LO
NG":GOTO8:ELSERETURN
110 'ee Codigos de maquina + usados ee
111 P=17:GOSUB2:P=E1:GOSUB2:P=D1:GOTO2 'LD DE,E1,D1
112 P=33:GOSUB2:P=E1:GOSUB2:P=D1:GOTO2 'LD HL,E1,D1
113 P=25:GOTO2 'ADD HL,DE
114 P=41:GOSUB2:P=41:GOTO2 '2x ADD HL,HL
115 P=235:GOTO2 'EX DE,HL
116 P=54:GOSUB2:P=C1:GOTO2 'LD (HL),C1
117 C1=VT+VO+VO:GOSUB83:P=42:GOSUB2:P=E1:GOSUB2:P=D1:GOTO2 'LD H
L,C1
118 P=42:GOSUB2:GOSUB84:GOSUB2:P=P1:GOTO2 'LD HL,(P1P)
119 P=34:GOSUB2:GOSUB84:GOSUB2:P=P1:GOTO2 'LD (P1P),HL
120 P=195:GOSUB2:P=E1:GOSUB2:P=D1:GOTO2 'JP E1D1
121 P=183:GOSUB2:P=237:GOSUB2:P=82:GOTO2 'OR A, SBC HL,DE
122 P=40:GOSUB2:P=3:GOTO2 'JR Z,3
123 P=225:GOTO2 'POP HL
124 P=229:GOTO2 'PUSH HL
125 P=209:GOTO2 'POP DE
126 P=213:GOTO2 'PUSH DE
127 P=205:GOSUB2:P=177:GOSUB2:P=9:GOTO2 '(4121H)=variavel
128 P=205:GOSUB2:P=164:GOSUB2:P=9:GOTO2 '(Stack)=(4121H)
129 P=193:GOSUB2:GOTO125 'POP BC, POP DE
130 'Rotinas Aritmeticas e Funcoes
131 P=205:GOSUB2:P=22:GOSUB2:P=7:GOTO2
132 P=205:GOSUB2:P=19:GOSUB2:P=7:GOTO2
133 P=205:GOSUB2:P=71:GOSUB2:P=8:GOTO2
134 P=205:GOSUB2:P=162:GOSUB2:P=8:GOTO2
135 P=205:GOSUB2:P=247:GOSUB2:P=19:GOTO2
136 P=205:GOSUB2:P=12:GOSUB2:P=1D:GOTO2
137 P=205:GOSUB2:P=231:GOSUB2:P=19:GOTO2
138 P=205:GOSUB2:P=239:GOSUB2:P=1D:GOSUB2:P=205:GOSUB2:P=119:GOS
UB2:P=9:GOTO2
139 P=205:GOSUB2:P=9:GOSUB2:P=8:GOTO2
140 P=205:GOSUB2:P=57:GOSUB2:P=20:GOTO2
141 P=205:GOSUB2:P=65:GOSUB2:P=21:GOTO2
142 P=205:GOSUB2:P=71:GOSUB2:P=21:GOTO2
143 P=205:GOSUB2:P=168:GOSUB2:P=21:GOTO2
144 P=205:GOSUB2:P=189:GOSUB2:P=21:GOTO2
145 P=205:GOSUB2:P=27:GOSUB2:P=21:GOTO2
146 P=62:GOSUB2:P=4:GOSUB2:P=50:GOSUB2:P=175:GOSUB2:P=64:GOTO2
147 P=205:GOSUB2:P=203:GOSUB2:P=9:GOTO2
148 'ee Entrada principal
149 CLEAR80:RESTORE:DEFINTA-0,8-X,Z:PRINT80,"MICRO BASIC COMPILE
R";TAB(4D)"NEWDOS2.D
"TAB(25)"BY Zorro /84

Program lines: 1000 a 9999":FORPC=1TO300:NEXTPC:PRINT"
Para instrucoes, leia Micro Sistemas":FORP=1TO30D:NEXTP:PRINT
150 M=0:@=0:IPN=0:IC=0:C1=0:V1=0:L=0:J=0:E1=0:D1=0:DH=0:DN=0
151 DEF FNA(Y)=Y+(Y)32767)*65536
152 INPUT"NUMERO DE BYTES P/ PROGRAMA (TOP MEM)";ZI:INPUT"NUMERO
 DE BASIC LINES MAXIMO A COMPILAR";@:INPUT"NUM. MAX. DE GOTO'S+@
OSUB'S";K
153 DIML1(@),A(K),L2(@),D(25),E(25),SM(10):T$=" : "
154 MR=0:INPUT"MEMORY OFFSET";I:INPUT"BYTES PARA TEXTO";PA:PB=-P
A-300-I:RR=65536
155 K=0:I=PEEK(16548)+256*PEEK(16549)+MS=-ZI+PB-MR+16384:PRINT"P
rocurando a linha )=1000...";!@=FNA(Y)
156 GOSUB10:PRINT@800,I;:IFI)=1000THEN157ELSE@=FNA(PEEK(@)+256*P
EEK(FNA(@+1))):GOTO156
157 PRINT"OK":PRINT"Entry point:";RR+MS;:INPUT"OK (S/N)";C$:IFC$
="N"THENINPUT"NOVO ENTRY POINT (dec)";I:I=I-RR+MS-I:IFI(=0THEN15
7ELSEMS=MS-I:PB=PB-I
158 PT=PB:MC=MS:M=M:INPUT"SINGLE PRECISION: A,AO,...Ax-1,...A9;
...;Z,ZO,...,Z9(X(=1D)";IS:INPUT"1-D ARRAY DIMENSION A(0-x). X)
=0";DO:INPUT"NUMERO DE POSSIVEIS 1-D VAR ARRAYS (A,B,...x).X(=26
.";NO
159 INPUT"2-D ARRAY SQUARE DIMENSION A(0-x,0-x). X)=0.";DT:INPUT
'NUMERO DE 2-D SQUARE ARRAYS (A,B...x). X(=26.";NT:INPUT"NUMERO
DE VARIAVEIS STRING (A$,B$,...Z$) ((=26).";NS:INPUT"STRING-VAR L
ENGTH. ((256).";SL
160 VT=-2*26+MS:VF=-4*26*(1+IS)+VT:VA=-4*NO*DO+VF:VD=-4*NT*DT*DT
-2*NT*DT+VA:VS=-NS*(SL+1)+VD:VN=-(SL+4)+VS
161 CLS:PRINTSTRING$(64,179);:PRINT@25,"ee RELATORIO ee":PRINT"N
umeros SPV adicionais:";TAB(32)IS;"
Dim de 1-D arrays=";TAB(32)DO;"
Dim de 2-D arrays=";TAB(32)DT;"
String length=";TAB(32)SL
162 PRINT"No de var. 1-Dim=";TAB(32)NO;"
Variaveis 2-Dim=";TAB(32)NT;"
Variaveis strings=";TAB(32)NS;"
Textos, area & start=";TAB(32)PA;" & ";PT
163 PRINT"Start of machine code=";TAB(32)MC;"
Integer & single var.=";TAB(32)VT;"&";VF;"
Start of 1-D & 2-D arrays=";TAB(32)VA;"&";VD;"
Start of string stor.=";TAB(32)VS;"
Temporary storage=";TAB(32)VN
164 IF65536+VN(=256*PEEK(16562)+PEEK(16561)THENPRINT"MEMORIA RES
ERVADA OVERLAP BASIC":END
165 PRINT@960,"APERTE...";:IFINKEY$=""THEN165ELSECLS:POKE16916,1
:PRINTSTRING$(64,95);:PRINT@0,"( ZORRO BASIC COMPILER )";
166 PRINT:PRINT:PRINT"Subrotinas:"PRINT"(";M;")..";
167 C1=M+3:GOSUB83:XH=D1:XL=E1:C1=M+37:GOSUB83:GOSUB120
168 DATA42,32,64,54,63,35,54,32,35,34,32,64,33,232,65,6,240,205,
217,5,245,72,6,0,9,54,0,33,232,65,241,216,175,201
169 FORI=1TO34:READP:GOSUB2:NEXTI
170 IFNT)0THENGOSUB363ELSEGOSUB372
171 GOSUB9
172 L=L+1:M1=FNA(PEEK(@)+PEEK(@+1)*256):L1(L)=PEEK(@+2)+PEEK(@+3
)*256
173 IFL1(L))9999THENL=L-1:GOTO201
174 PRINT@960,"#";L1(L);"(";M;")..";:L2(L)=M1@=@+4
175 C=0:GOSUB3:IFC=2THEN200
176 IFPC=1330RPC=1340RPC=136OR(PC)137ANDPC(14D)ORPC=142ORPC=1440
RPC=148OR(PC)149ANDPC(176ANDPC()16D)OR(PC)178ANDPC(189)OR(PC)189
ANDPC(202ANDPC()193)ORPC=203ORPC=204ORPC=210ORPC=211THEN8 'erro
177 IFPC=215OR(PC)216ANDPC(221)OR(PC)228ANDPC(251)THEN8 'erro
178 IFPC=140THENGOSUB3ELSEIFPC=16OTHENPRINT" Out!";:GOSUB374:GOS
UB379:GOTO195 'Out
179 IFPC)64ANDPC(91ANDPN=37@=@-1:PRINT" Let%";:GOSUB213:GOTO195
 'integer let
180 'Simples Precisao Let
181 IFPC)64ANDPC(91ANDPN()37ANDPN()36@=@-1:PRINT" Let!";:GOSUB1
2:GOSUB3:GOSUB3:IF(PC=49ANDPN=207)OR(PC=48ANDPN=205)GOSUB309:GOT
0195ELSE@=@-2:GOSUB3:IFPC()213THEN8ELSEGOSUB28:GOSUB20:GOSUB147:
GOTO195
182 IFPC)64ANDPC(91ANDPN=36@=@-1:PRINT" Let$";:GOSUB314:GOTO195
 'string let
183 IFPC=178THENPRINT" Print";:GOSUB232:GOTO195ELSEIFPC=176THEN
PRINT" Def";:GOSUB381:GOSUB379:GOTO195 'print, def
184 IFPC=141THENPRINT" Goto";:GOSUB303:GOTO195ELSEIFPC=193THENP
RINT" Gos";:GOSUB385:GOSUB379:GOTO195 'goto, gosub
185 IFPC=143PRINT" If";:GOSUB263:GOTO195 'if..then..else
186 IFPC=145PRINT" Gosub";:GOSUB305:GOTO195 'gosub
187 IFPC=146PRINT" Ret";:GOSUB307:GOTO195 'return
188 IFPC=132THENPRINT" Cls";:P=205:GOSUB2:P=201:GOSUB2:P=1:GOSU
B2:GOSUB3:IFPC()58ANDC()2THEN8ELSE195 'cls
189 IFPC=137PRINT" Input";:GOSUB254:GOSUB379:GOTO195 'input
190 IFPC=129PRINT" For";:GOSUB332:GOTO195 'for
191 IFPC=135PRINT" Next";:GOSUB342:GOTO195 'next
192 IFPC=130ORPC=131PRINT" Set & Res";:GOSUB344:GOTO195 'set &
reset
193 IFPC=177PRINT" Poke";:GOSUB357:GOTO195 'poke
194 IFPC=128PRINT" End";:P=205:GOSUB2:P=157:GOSUB2:P=1D:GOSUB2:
P=201:GOSUB2:GOSUB3 'end
195 C1=PEEK(@-1):IFFO()OTHENI=196ELSEIFC1=58THENPRINTT$;:GOTO175EL
SEIFC1=149THENPRINT" Else";:GOTO175ELSE200
196 IFC1=58THENPRINTT$;:GOTO175ELSEIFC1=149THENC1=M:GOSUB79:PRIN
T" Else";:I=SM(FO):FD=FD-1:IFPEEK(I)=4DTHENI=I+3:GOTO197ELSEI=I+
1:GOSUB19B:IFPEEK(I+2)=250ORPEEK(I+2)=242THENI=I+3:GOTO197ELSE17
5ELSEIFC1()OTHENPRINT" Then1";:@=@-1:GOTO175ELSE199
197 GOSUB198:GOTO175
198 POKEI,E1:POKEI+1,D1:RETURN
199 IFFO()OTHENC1=M:GOSUB79:FORV1=1TOFO:I=SM(V1):IFPEEK(I)=4DTHE
NI=I+3:GOSUB19B:NEXTV1ELSEI=I+1:GOSUB19B:IFPEEK(I+2)=250ORPEEK(I
+2)=242THENI=I+3:GOSUB19B:NEXTV1ELSENEXTV1 'Ajuste de Else
200 FO=DI@=M1:PRINT:GOTO171 'proxima linha
201 PRINT:GOSUB9:PRINT@960,"AJUSTANDO JUMP ADDRESSES . . .";:IFK
=0GOTO206
202 FORI=1TOK:DN=PEEK(A(I))+256*PEEK(A(I)+1):DH=0
203 FORJ=1TOL1:IFDN=L1(J)THENDH=L2(J):PRINTL1(J);
204 NEXTJ:C1=DH:GOSUB83:POKEA(I),E1:POKEA(I)+1,D1:NEXTI
205 'Finalizacao ee
206 PRINT:DEFUSR0=MC:R1=RR+MC:R2=RR+1+H:R3=RR+VN:R4=RR+PB 'prepa
ra Defusr0 p/ execucao
207 PRINT:PRINT"Inicio da area das variaveis...";TAB(32)R3:PRINT
"Inicio do programa...";TAB(32)R1:PRINT"Fim do programa...";TAB(
32)R2:PRINT"Fim da area de textos...";TAB(32)R4
208 IFN)PTTHENPRINT"ERRO: PROGRAMA E TEXTO OVERLAP"
209 PRINT"<S) TO SAVE OR (R) TO RUN MACHINE CODE.....
210 A$=INKEY$:IFA$=""THEN210ELSEIFA$="R"THENCLS:POKE16916,0:X=US
R(0):GOTO209
211 IFA$()"S"THEN209ELSEPRINT:PRINT"Para gravar, execute em Disk
-Basic modo diretot
CMD"DUMP,F(lename,"R1;",",R4;",",R1:PRINT:PRINT"Inicio da area
de memoria reservada para as variaveis =";R3;"
(TraDos Basic: Memory Size ou NewDos Basic: Himem)":END
212 'Operacoes com inteiros
213 GOSUB3:GOSUB5:GOSUB6:V1=PC-65:GOSUB3:IFPC()213THEN8
214 GOSUB3:IF(PC=49ANDPN=207)OR(PC=48ANDPN=205)THEN227
215 IFPC=216GOSUB3:GOSUB28:GOSUB146:P=205:GOSUB2:P=61:GOSUB2:P=1
:GOSUB2:IFPC()41THEN8ELSEGOSUB3:GOTO119 'int
216 IFPC()229THEN218ELSEV3=V1:GOSUB3:GOSUB3:GOSUB77:IFC1=-1THENV
1=PC-65:GOSUB5:GOSUB6:GOSUB3:GOSUB11B:P=126:GOSUB2ELSEP=58:GOSUB
2:P=E1:GOSUB2:P=D1:GOSUB2
217 P=36:GOSUB2:P=0:GOSUB2:P=111:GOSUB2:V1=V3:GOSUB119:GOTO3 'pe
ek
218 IFPC=198THENV3=V1:GOSUB344:P=42:GOSUB2:P=33:GOSUB2:P=65:GOSU
B2:V1=V3:GOTO119ELSEIFPC=245ANDPN=40THEN400 'point
219 V2=V1:@=@-1:GOSUB3:IFPC=206ANDPN)47ANDPN(58GOSUB77:GOSUB112:
GOTO221ELSEIFPC=206THENE1=D1:D1=0:GOSUB112:GOTO221
220 GOSUB77:IFC1()-1GOSUB112ELSEV1=PC-65:GOSUB5:GOSUB6:GOSUB118:
GOSUB3
221 IFPC=58ORC=2V1=V2:GOTO119
222 IFPC=20&ANDPN)47ANDPN(58&G=205ELSESG=PC:GOSUB3
223 GOSUB225:IFSG=205DOSUB113ELSEGOSUB121
224 GOTO221
225 GOSUB77:IFC1()-1GOTO111ELSEV1=PC-65:GOSUB5:GOSUB6:P=237:GOSU
B2:P=V1:GOSUB2:GOSUB84:GOSUB2:P=P1:GOSUB2:GOTO3
226 'Transferencia de variaveis inteiras, basic X usr
227 IFPC=49ANDPN=207CM=1ELSECM=0
228 @=@+1:GOSUB3:MF=VN:GOSUB10D:@=@+1:GOSUB112:P=205:GOSUB2:P=13
:GOSUB2:P=3B:GOSUB2
229 IFCM=0P=26:GOSUB2:P=111:GOSUB2:P=19:GOSUB2:P=26:GOSUB2:P=103
:GOSUB2:GOTO119 'Basic p/ Z80
230 GOSUB118:P=125:GOSUB2:P=18:GOSUB2:P=19:GOSUB2:P=124:GOSUB2:P
=18:GOTO2 'Z80 p/ Basic
231 'Print
232 GOSUB3:IFPC=58ORC=2P1=13:GOTO251
233 IFPC()64ANDPC()96THEN244ELSEGOSUB3
234 'Print@
235 IFPC(58@=@-1:GOSUB24D:C1=C1+153&D:GOSUB3:GOSUB79:GOSUB112ELS
EV1=PC-65:GOSUB5:GOSUB6:GOSUB3:GOSUB3:GOSUB118:D1=6D:E1=D:GOSUB1
11:GOSUB113
236 IFPC()247P=34:GOSUB2:P=32:GOSUB2:P=64:GOSUB2:GOTO244ELSEGOSU
B3:IFPC()40THENBELSEGOSUB240:GOSUB116
237 GOSUB3:IFPC=205GOSUB3:IFPC()247THENBELSEP=35:GOSUB2:GOTO236
238 IFPC=59GOSUB3
239 IFPC()58ANDC()2THEN8ELSERETURN
240 C$=""
241 GOSUB3:IFPC()41ANDPC()44THENC$=C$+CHR$(PC):IFPC(480RPC)57THE
N8ELSEGOTO241
242 C1=VAL(C$):RETURN
243 'Expressao Simples Precisao & Print String
244 IFPC)64ANDPC(91ANDPN=36@=@+1:V1=PC-65:C1=V5+V1+(SL+1):GOSUB2
52:GOTO247
245 IFPC=34THENGOSUB39D:GOTO247
246 @=@-1:GOSUB28:GOSUB146:P=205:GOSUB2:P=189:GOSUB2:P=15:GOSUB2
:P=62:GOSUB2:P=3:GOSUB2:P=18:GOSUB2:GOSUB145 'print simples prec
isao
247 IFPC=44THEN8
248 IFPC=59GOSUB3:IFPC()58ANDC()2THEN244ELSERETURN
249 IFPC=58ORC=2P1=13:GOTO251
250 GOTO8
251 P=62:GOSUB2:P=P1:GOSUB2:P=205:GOSUB2:P=58:GOSUB2:P=3:GOTO2 '
scroll
252 GOSUB83:GOSUB112:GOSUB145:GOSUB3:IFPC=34@OTO3ELSERETURN 'pri
nt string
253 'INPUT
254 GOSUB3:IFPC=34THENGOSUB39O:IFPC()59THEN8ELSE254ELSEGOSUB5
255 P=205:GOSUB2:P=XL:GOSUB2:P=XH:GOSUB2:IFPN()36THEN259
256 V1=PC-65:C1=V5+V1+(SL+1):GOSUB83:GOSUB111
257 P=126:GOSUB2:P=183:GOSUB2:P=40:GOSUB2:P=5:GOSUB2:P=18:GOSUB2
:P=35:GOSUB2:P=19:GOSUB2:P=24:GOSUB2:P=247:GOSUB2:P=62:GOSUB2:P=
3:GOSUB2:P=18:GOSUB2
258 @=@+1:GOTO3
259 P=205:GOSUB2:P=108:GOSUB2:P=14:GOSUB2
260 IFPN=37P=205:GOSUB2:P=127:GOSUB2:P=1D:GOSUB2:V1=PC-65:GOSUB8
4:E1=P:D1=P1:GOSUB112:P=237:GOSUB2:P=75:GOSUB2:P=33:GOSUB2:P=65:
GOSUB2:P=113:GOSUB2:P=35:GOSUB2:P=112:GOSUB2:GOSUB3:GOTO3
261 @=@-1:CF=1:GOSUB12:CF=0:P=58:GOSUB2:P=175:GOSUB2:P=64:GOSUB2
:P=222:GOSUB2:P=4:GOSUB2:GOSUB122:P=205:GOSUB2:P=2D4:GOSUB2:P=10
:GOSUB2:GOSUB2D:GOSUB147:GOTO3
262 'Integer If-Then Rotina
263 GOSUB3:IFPN()37THEN279ELSEGOSUB6:V1=PC-65:GOSUB11B:GOSUB115:
GOSUB3
264 IFPC=212ANDPN=213ORPC=213ANDPN=212W1=1:@=@+1:GOTO27D
265 IFPC=214ANDPN=213ORPC=213ANDPN=214W1=2:@=@+1:GOTO27D
266 IFPC=212ANDPN=214ORPC=214ANDPN=212W1=3:@=@+1:GOTO27D
267 IFPC=212W1=5
268 IFPC=214W1=4
269 IFPC=213W1=6
270 GOSUB3:IFPC(58ORPC=206GOSUB77:GOSUB112ELSEGOSUB5:GOSUB6:V1=P
C-65:GOSUB11B:GOSUB3
271 IF(PC=202ANDPC(58ANDPN)47)ORPC=141THENGOSUB3:PRINT" Then#";
ELSEIFPC=202ANDPN=141THENGOSUB3:GOSUB3ELSEIFPC=202THENGOSUB3:GOT
O275ELSE8
272 GOSUB77:GOSUB7
273 GOSUB274:I=0:GOTO289
274 P=205:GOSUB2:P=57:GOSUB2:P=10:GOTO2
275 GOSUB274
276 FO=FO+1:SM(FO)=M:IFW1(4THENW1=W1+3ELSEW1=W1-3
277 I=1:GOTO289
278 'Single Precision If-Then rotina
279 @=@-1:GOSUB28:GOSUB128
280 IFPC=212ANDPN=213ORPC=213ANDPN=212W1=1:@=@+1:GOTO286
281 IFPC=214ANDPN=213ORPC=213ANDPN=214W1=2:@=@+1:GOTO286
282 IFPC=212ANDPN=214ORPC=214ANDPN=212W1=3:@=@+1:GOTO286
283 IFPC=212W1=5
284 IFPC=214W1=4
285 IFPC=213W1=6
286 GOSUB28:GOSUB129:GOSUB136
287 IF(PC=202ANDPN(58ANDPN)47)ORPC=141THENGOSUB3:PRINT" Then#";
ELSEIFPC=202ANDPN=141THENGOSUB3:GOSUB3ELSEIFPC=202THENGOSUB3:GOT
O27&ELSE8
288 I=0:GOSUB77:GOSUB7
289 D=D1:E=E1:ONW1GOTO290,291,292,294,293,295
290 GOSUB296:GOTO299
291 GOSUB296:GOTO297
292 GOTO298
293 P=40:GOSUB2:P=3:GOSUB2:GOTO299
294 P=40:GOSUB2:P=3:GOSUB2:GOTO297
295 GOTO296
296 P=202:GOTO3D1
297 P=242:GOTO3D1
298 P=194:GOTO3D1
299 P=250:GOTO3D1
300 I=0:P=195:GOTO3D1
301 GOSUB82:IFI=1THENP=0:GOSUB2:GOTO2ELSEK=K+1:A(K)=M:P=E:GOSUB2:
P=D:GOTO2
302 'Goto
303 GOSUB3:GOSUB77:GOSUB7:D=D1:E=E1:GOTO300
304 'Gosub
305 GOSUB3:GOSUB77:GOSUB7:D=D1:E=E1:C1=M+7:GOSUB83:GOSUB111:GOSU
B126:GOTO300
306 'Return
307 GOSUB123:P=233:GOTO2
308 'Transferencia Var. Simples Precisao
309 IFPC=49ANDPN=207THENCM=1ELSECM=0
310 GOSUB3:GOSUB3:MF=VN:GOSUB10D:GOSUB3:GOSUB112:P=205:GOSUB2:P=
13:GOSUB2:P=3B:GOSUB2
311 IFCM=0GOSUB115:GOSUB127:GOSUB2D:GOTO147
312 GOSUB126:GOSUB2D:GOSUB127:GOSUB123:GOTO147
313 'Operacao com Strings
314 GOSUB3:V1=PC-65:MF=VS+V1+(SL+1)
315 GOSUB3:IFPC()36THEN8
316 GOSUB3:IFPC()213THEN8
317 GOSUB3:IFPC=247THEN322ELSEIFPC)64ANDPC(91ANDPN=36THEN394ELSE
IFPC()34THEN8
318 PC=PEEK(@):@=@+1:FP=1:GOSUB1DD:FP=0
319 GOSUB3:IFPC=34THEN3ELSERETURN
321 'Strings com Chr$
322 C1=MF:GOSUB83:GOSUB112
323 GOSUB3:IFPC()4OTHEN8ELSEC$=""
324 GOSUB3:IFPC)64ANDPC(91THENV1=PC-65:GOSUB6:GOSUB3:IFPC()41THE
N8ELSEP=58:GOSUB2:GOSUB84:GOSUB2:P=P1:GOSUB2:P=119:GOSUB2:GOTO32
9
325 IF(PC(48ORPC)57)ANDPC()41THEN8
326 IFPC()41THENC$=C$+CHR$(PC):GOSUB3:GOTO325
327 C1=VAL(C$):GOSUB116
329 GOSUB3:IFPC=205GOSUB3:IFPC()247THEN8ELSEP=35:GOSUB2:GOTO323
330 IFPC=58ORC=2THENP=35:GOSUB2:C1=3:GOTO116ELSE8
331 'For Rotina
332 C1=M+7:GOSUB83:GOSUB3:GOSUB5:GOSUB6:V1=PC-65:GOSUB84:GOSUB3:
D(V1)=D1:E(V1)=E1:IFPC()213THEN8
333 GOSUB3:IFPC(58ORPC=206GOSUB77:J1=0:ID=D1:IE=E1:ELSEJ1=1:V2=P
C-65:GOSUB6:C1=VT+V2+2:GOSUB83:ID=D1:IE=E1:GOSUB3
334 IFPC()189THEN8
335 GOSUB3:IFPC(58ORPC=206GOSUB77:J2=0:FD=D1:FE=E1ELSEJ2=1:V3=PC
-65:GOSUB6:C1=VT+V3+2:GOSUB83:FD=D1:FE=E1:@=@+1
336 IFJ2=0THENP=33ELSEP=42
337 GOSUB2:P=FE:GOSUB2:P=FD:GOSUB2:GOSUB124
338 IFJ1=0THENE1=IE:D1=ID:GOSUB112
339 IFJ1=1THENP=42:GOSUB2:P=IE:GOSUB2:P=ID:GOSUB2
340 GOSUB119:IFPEEK(@-1)()5BANDPEEK(@-1)()0THEN8ELSERETURN
341 'Next Rotina
342 GOSUB3:GOSUB5:GOSUB6:V1=PC-65:GOSUB118:GOSUB125:GOSUB126:GOS
UB124:GOSUB121:GOSUB123:P=35:GOSUB2:P=194:GOSUB2:P=E(V1):GOSUB2:
P=D(V1):GOSUB2:GOSUB123:GOTO3
343 'Rotinas Point, Set & Reset
344 IFPC=130THENW=1ELSEIFPC=131THENW=128ELSEIFPC=190THENW=0
345 MA=M
346 GOSUB3:IFPC()4OTHEN8ELSEGOSUB3:GOSUB77:IFC1=-1GOSUB5:GOSUB6:
V1=PC-65:GOSUB84:D2=P1:E2=P:C2=1:ELSEE2=E1:C2=0:IFPC()44THEN8
347 IFC2=1GOSUB3:IFPC()44THEN8
348 GOSUB3:GOSUB77:IFC1=-1GOSUB5:GOSUB6:V1=PC-65:GOSUB84:D3=P1:E
3=P:C3=1:ELSEE3=E1:C3=0:IFPC()41THEN8
349 IFC3=1GOSUB3:IFPC()41THEN8
350 GOSUB3:IFPC()58ANDC()2THEN8
351 C1=MA+18+C2+C3:GOSUB83:GOSUB112:GOSUB124:E1=126:D1=7:GOSUB11
2:P=62:GOSUB2:P=W:GOSUB2:P=245:GOSUB2
352 IFC2=1THENP=58:GOSUB2:P=E2:GOSUB2:P=D2:GOSUB2:ELSEP=62:GOSUB
2:P=E2:GOSUB2
353 P=245:GOSUB2
354 IFC3=1THENP=58:GOSUB2:P=E3:GOSUB2:P=D3:GOSUB2ELSEP=62:GOSUB2
:P=E3:GOSUB2
355 E1=80:D1=1:GOTO120
356 'Rotina Poke
357 GOSUB3:GOSUB77:IFC1=-1THENGOSUB5:GOSUB6:V1=PC-65:GOSUB118:GO
SUB3ELSEGOSUB112
358 IFPC()44THEN8
359 GOSUB3:GOSUB77:IFC1=-1GOSUB5:GOSUB6:V1=PC-65:GOSUB84:E1=P:D1
=P1:P=58:GOSUB2:P=E1:GOSUB2:P=D1:GOSUB2:GOSUB3ELSEP=62:GOSUB2:P=
E1:GOSUB2
360 IFPC()58ANDC()2THEN8
361 P=119:GOTO2
362 '2-D code store
363 PRINT:PRINT:PRINT"Rotina p/armazenar matrizes 2-D:":PRINT:C1
=VD:GOSUB83:GOSUB112
364 C1=VD+4+NT*DT+DT:GOSUB83:P=221:GOSUB2:GOSUB112:C1=4+DT:GOSUB
79:GOSUB111
365 C1=NT*DT:GOSUB79:P=1:GOSUB2:P=E1:GOSUB2:P=D1:GOSUB2
366 P=221:GOSUB2:P=117:GOSUB2:P=0:GOSUB2:P=221:GOSUB2:P=35:GOSUB
2:P=221:GOSUB2:P=116:GOSUB2:P=0:GOSUB2:P=221:GOSUB2:P=35:GOSUB2:
GOSUB113:P=13:GOSUB2
370 C1=M-12:GOSUB83:P=194:GOSUB2:P=E1:GOSUB2:P=D1:GOSUB2
371 P=5:GOSUB2:P=14:GOSUB2:P=255:GOSUB2:C1=M-18:GOSUB83:P=242:GO
SUB2:P=E1:GOSUB2:P=D1:GOSUB2
372 PRINT@840,"Compilacao principal:"RETURN
373 'Rotina Out
374 GOSUB3:GOSUB77:IFC1=-1THEN8ELSEP0=C1
375 IFP0(DORP0)255THEN8
376 IFPC()44THEN8
377 GOSUB3:GOSUB77:IFC1=-1THENGOSUB5:GOSUB6:V1=PC-65:GOSUB84:E1=
P:D1=P1:P=58:GOSUB2:P=E1:GOSUB2:P=D1:GOSUB2:GOSUB3ELSEP=62:GOSUB
2:P=E1:GOSUB2
378 P=211:GOSUB2:P=P0:GOTO2
379 IFPC()58ANDC()2THEN8ELSERETURN
380 'DefUsr
381 GOSUB3:IFPC=193ANDPN=213THEN@=@+1ELSE8
382 GOSUB3:GOSUB77:IFC1=-1THENGOSUB5:GOSUB6:V1=PC-65:GOSUB118:GO
SUB3ELSEGOSUB112
383 P=34:GOSUB2:P=142:GOSUB2:P=64:GOTO2
384 'Rotina Usr
385 GOSUB3:IFPC()4OTHEN8
386 GOSUB3:GOSUB77:IFC1=-1THENGOSUB5:GOSUB6:V1=PC-65:GOSUB84:E1=
P:D1=P1:P=58:GOSUB2:P=E1:GOSUB2:P=D1:GOSUB2:GOSUB3ELSEP=62:GOSUB
2:P=E1:GOSUB2
387 IFPC()41ORC1(D0RC1)255THEN8ELSEGOSUB3
388 P=205:GOSUB2:P=PEEK(16526):GOSUB2:P=PEEK(16527):GOTO2
389 'Armazenamento de textos
```

```
390 C1=PB:PRINT" (";PB;")";:IFPB)-30OTHENPRINT"TEXTO OVERFLOW":G
0T08
392 GOSUB84:IFPC()34THENPRINTCHR$(PC);:POKEPB,PC:PB=PB+1:GOTO392
393 POKEPB,3:PB=PB+1:GOTO252
394 C1=MF:GOSUB83:GOSUB111:V1=PC-65:C1=VS+V1*(SL+1):GOSUB83:GOSU
B112
397 P=1:GOSUB82:P=SL+1:GOSUB82:P=0:GOSUB82:P=237:GOSUB82:P=176:GOSUB
82:@=@+1:GOSUB83:GOTO379
399 'Val
400 @=@+1:GOSUB83:IFPC(65ORPC)90ORPN()36THENBELSEV2=PC-65:C1=VS+(
SL+1)*V2:GOSUB83:GOSUB83
401 IFPC()41THENBELSEGOSUB83:GOSUB112:P=205:GOSUB82:P=90:GOSUB82:P
=30:GOSUB82:@=@+1:C1=VT+2*V1:GOSUB83:GOSUB112:P=115:GOSUB82:P=35:G
OSUB82:P=114:GOTO2
500 '*** Compiler/Bas ***
```

QUASI/USR

Este é o primeiro módulo Z80 a ser compilado para a formação de Quasar IV. A listagem 2 apresenta o programa já escrito na forma que o compilador gosta. Vamos chamá-lo de Quasl/CMP.

Para evitar confusão, deve-se dar os seguintes parâmetros de entrada para a compilação na ordem em que são solicitados: 5000, 150, 100, 5500 (offset para QuasII/USR, 100 (texto), S (deve ser 54636), 1 (SPVs), 0, 0, 0, 0, 0 (SVs), 0.

É IMPORTANTE não errar nem mudar estes dados, pois o Entry-Point resultante é utilizado na chamada USR de Quasar IV. E se mudar aqui, vai ter que mudar lá, e isso é válido para o Quasl e o QuasII.

Não se pode esquecer de SALVAR o resultado, dando um DUMP com os parâmetros fornecidos pelo compilador e com o FILENAME da chamada de Quasar IV, que é QUASI/USR.

Se até aqui foi tudo bem, ótimo. Caso contrário, é bom esfriar um pouco a cabeça e retroceder a leitura.

Listagem 2 - Quasl/CMP

```
1000 'QUASI/CMP FOR COMPILER ******** Zorro/84 **********
1005 GOTO1440
1010 'MOVE 16 PONTOS
1015 A=EX:BX=INT(B):RESET(AX,BX):AX=AX+5:IFAX)126THEN1020ELSESET
(AX,BX):GOTO1025
1020 AX=XX:B=YX
1025 BX=INT(P):EX=INT(D):RESET(BX,EX):BX=BX+5
1030 IFBX)126THEN1035ELSESET(BX,EX):P=BX:GOTO1040
1035 P=XX:D=YX
1040 RESET(CX,DX):CX=CX+2:DX=DX-2:IFCX)126THEN1050ELSEIFDX(1THEN
1050
1045 SET(CX,DX):GOTO1055
1050 CX=XX:DX=YX
1055 BX=INT(E):RESET(BX,FX):FX=FX-3:IFFX(1THEN1060ELSESET(BX,FX)
:GOTO1065
1060 FX=YX:E=XX
1065 BX=INT(G):EX=INT(H):RESET(BX,EX):EX=EX-3
1070 IFEX(1THEN1075ELSESET(BX,EX):H=EX:GOTO1080
1075 G=XX:H=YX
1080 RESET(GX,HX):GX=GX-2:HX=HX-2:IFGX(1THEN1090ELSEIFHX(1THEN1O
90
1085 SET(GX,HX):GOTO1095
1090 G=XX:HX=YX
1095 BX=INT(J):RESET(IX,BX):IX=IX-5:IFIX(1THEN1100ELSESET(IX,BX)
:GOTO1105
1100 IX=XX:J=YX
1105 BX=INT(Q):EX=INT(R):RESET(BX,EX):BX=BX-5
1110 IFBX(1THEN1115ELSESET(BX,EX):Q=BX:GOTO1120
1115 Q=XX:R=YX
1120 RESET(KX,LX):KX=KX-2:LX=LX+2:IFKX(1THEN1130ELSEIFLX)46THEN1
130
1125 SET(KX,LX):GOTO1135
1130 KX=XX:LX=YX
1135 BX=INT(M):RESET(BX,MX):NX=NX+3:IFNX)46THEN1140ELSESET(BX,NX
):GOTO1145
1140 NX=YX:M=XX
1145 BX=INT(W):EX=INT(Z):RESET(BX,EX):EX=EX+3
1150 IFEX)46THEN1155ELSESET(BX,EX):Z=EX:GOTO1160
1155 W=XX:Z=YX
1160 SET(XX,YX):RESET(OX,PX):OX=OX+2:PX=PX+2:IFOX)126THEN1170ELS
EIFPX)46THEN1170
1165 SET(OX,PX):GOTO1175
1170 OX=XX:PX=YX
1175 RESET(QX,RX):QX=QX+4:RX=RX-1:IFQX)126THEN1185ELSEIFRX(1THEN
1185
1180 SET(QX,RX):GOTO1190
1185 QX=XX:RX=YX
1190 RESET(SX,TX):SX=SX-4:TX=TX-1:IFSX(1THEN1200ELSEIFTX(1THEN12
00
1195 SET(SX,TX):GOTO1205
1200 SX=XX:TX=YX
1205 RESET(UX,VX):UX=UX-4:VX=VX+1:IFUX(1THEN1215ELSEIFVX)46THEN1
215
1210 SET(UX,VX):GOTO1220
1215 UX=XX:VX=YX
1220 EX=INT(A):RESET(WX,ZX):WX=WX+4:ZX=ZX+1:IFWX)126THEN1230ELSE
IFZX)46THEN1230
1225 SET(WX,ZX):RETURN
1230 WX=XX:ZX=YX:RETURN
1235 'KEYBOARD MOVE
1240 JX=PEEK(14368):IFJX=16THEN1250ELSEIFJX=64THEN1255
1245 JX=PEEK(14344):IFJX=4THEN1260ELSEIFJX=1THEN1265ELSERETURN
1250 XX=XX+1:IFXX(127THEN1245ELSEXX=126:GOTO1245
1255 XX=XX-1:IFXX)=1THEN1245ELSEXX=1:GOTO1245
1260 YX=YX+1:IFYX(47THEN1270ELSEYX=46:RETURN
1265 YX=YX-1:IFYX)=1THEN1270ELSEYX=1:RETURN
1270 RETURN
1275 'RANDOM MOVE
1280 IFEX()0THEN1290ELSEEX=INT(10*RND(0)-4.5)
1285 IFMX()0THEN1295ELSEMX=INT(10*RND(0)-4.5):RETURN
1290 IFEX)0THEN1300ELSEIFEX(0THEN1305
1295 IFMX)0THEN1310ELSEGOTO1315
1300 XX=XX+1:EX=EX-1:IFXX(127THEN1285ELSEXX=126:GOTO1285
1305 XX=XX-1:EX=EX+1:IFXX)=1THEN1285ELSEXX=1:GOTO1285
1310 YX=YX+1:MX=MX-1:IFYX(47THEN1270ELSEYX=46:RETURN
1315 YX=YX-1:MX=MX+1:IFYX)=1THEN1270ELSEYX=1:RETURN
1320 'VISOR
1325 PRINT@412,CHR$(156);:PRINT@540,CHR$(180);
1330 PRINT@546,CHR$(184);:PRINT@418,CHR$(172);:RETURN
1335 'TESTA CENTRO
1340 IFXX)69THEN1355ELSEIFXX(56THEN1355
1345 IFYX)26THEN1355ELSEIFYX(19THEN1355
1350 C=0:RETURN
1355 C=C+1:IFC)10THEN1365ELSERETURN
1360 'IMPACTO
1365 C=0:CO=CO+1
1370 GOSUB1480
1375 SET(62,22)
1380 FORJX=1TO22:KX=INT(62-(2*JX)):LX=INT(2*JX+62):NX=22-JX:NX=2
2+JX
1385 FORBX=KXTOLX:SET(BX,MX):SET(BX,NX):NEXTBX
1390 FORBX=MXTONX:SET(KX,BX):SET(LX,BX):KX=KX+1:LX=LX-1:SET(KX,B
X):SET(LX,BX):KX=KX-1:LX=LX+1:NEXTBX
1395 GOSUB1485:GOSUB1015:GOSUB1480:NEXTJX
1400 GOSUB1425:CLS:GOSUB1425:CLS:GOSUB1425
1405 IFCO(6THEN1410ELSERETURN
1410 PRINT@473,"(* IMPACTO *)";:PRINT@965,"TELAS:";5-CO;:PRINT@1
010,"TIME:";TO;" ";
1415 FORJX=1TO512:BX=INT(959*RND(0)):PRINT@BX,CHR$(128);:NEXTJX
1420 CLS:GOSUB1485:GOSUB1470:RETURN
1425 FORJX=0TO1023:PRINT@JX,CHR$(191);:NEXTJX:RETURN
1430 IX=0:IX=1*I:END
1435 'MAIN LOOP
1440 CLS:XX=62:YX=22:TO=300:C=0:CO=0:GOSUB1470
1445 AX=XX:CX=XX:DX=YX:FX=YX:GX=XX:HX=YX:IX=XX:KX=XX:LX=YX:NX=YX
:OX=XX:PX=YX:QX=XX:RX=YX:SX=XX:TX=YX:UX=XX:VX=YX:WX=XX:ZX=YX:EX=
0:MX=0:B=YX:E=XX:H=XX:J=YX:P=XX-(XX/2):D=YX:G=XX:H=YX/2:Q=XX/2:R
=YX:W=XX:Z=YX-(YX/2)
1450 FORJX=1TO7:GOSUB1325:GOSUB1015:SET(XX,YX):NEXTJX:RESET(XX,Y
X)
1455 GOSUB1280:GOSUB1240:GOSUB1340:IFCO)5THEN1430
1460 TO=TO-1:PRINT@52,"Time:";TO;" ";:IFTO)0THEN1450ELSEIX=1:IX=
1*I:END
1465 'MOLDURA
1470 FORJX=0TO127:SET(JX,0):SET(JX,47):NEXTJX
1475 FORJX=0TO47:SET(0,JX):SET(127,JX):NEXTJX:RETURN
1480 K=KX:L=LX:MO=MX:M=NX:RETURN
1485 KX=INT(K):LX=INT(L):MX=INT(MO):NX=INT(N):RETURN
10000 END
```

QUASII/USR

É o segundo módulo Z80. A listagem 3 contém o programa para o compilador: QuasII/CMP. Os parâmetros para esta compilação são, na mesma ordem de entrada, os seguintes: 4100, 150, 70, 0 (offset), 900 (texto), S (deve ser 60236), 0, 0, 0, 0, 0, 1 (SVs) e 10.

No DUMP para salvar, deve-se dar o FILENAME de chamada de Quasar IV, que é QUASII/USR. Só isto.

Listagem 3 - QuasII/CMP

```
1000 'JERRY/CMP FOR QUASAR IV  (QUASII/USR BY COMPILER)  Zorro/8
4
1005 EX=0+I:CLS:TX=30:FORIX=1TO5:SX=30:GOSUB1290:SX=15:GOSUB1290
:NEXTIX
1010 FORIX=0TO127:SET(IX,0):SET(IX,1):SET(IX,42):SET(IX,43):NEXT
IX
1015 FORIX=0TO43:SET(0,IX):SET(1,IX):SET(126,IX):SET(127,IX):NEX
TIX
1020 FORIX=82TO85:FORJX=24TO25:SET(IX,JX):NEXTJX:NEXTIX
1025 FORIX=96TO99:FORJX=6TO7:SET(IX,JX):NEXTJX:NEXTIX
1030 FORJX=36TO40:SET(114,JX):SET(117,JX):NEXTJX:SET(116,40):SET
(115,40)
1035 SET(124,5):SET(122,4):SET(118,4):SET(118,2):SET(122,2):SET(
124,7):SET(120,7):POKE15484,145:POKE15549,145:POKE15547,145
1040 FORKX=1TO15:GOSUB1285:NX=INT(10*RND(0)+1)
1045 FORIX=1TONZ
```

DESCUBRA AS DIFERENÇAS

Aparentemente estes dois cabos são iguais.
Olhe bem e tente descobrir as diferenças.

Solução:

1 - O cabo de cima é AUDIOFLEX. Ele tem continuidade de características elétricas ao longo de toda linha, porque é fabricado com o melhor equipamento e sua qualidade é controlada em toda linha de fabricação.
2 - O cabo de cima é AUDIOFLEX. Ele é fabricado com cobre eletrolítico novo e polietileno novo - nada de matéria-prima recuperada.
3 - O cabo de cima é AUDIOFLEX. Sua montagem é rápida e fácil, devido às diversificações de tipos e cortes bobinados no comprimento exato.
4 - O cabo de cima é AUDIOFLEX. Ele passa pelo mais avançado controle de qualidade.
5 - O cabo de cima é AUDIOFLEX. Ele é fabricado por uma empresa que só fabrica cabos especiais.
6 - O cabo de cima é AUDIOFLEX. Ele é feito com a mais alta tecnologia.
7 - O cabo de cima é AUDIOFLEX. Ele é fabricado em mais de 18 tipos diferentes.
8 - O cabo de cima é AUDIOFLEX. A empresa que o fabrica tem um Departamento de Engenharia preparado para indicar qual o melhor tipo para seu caso.
9 - O cabo de cima é AUDIOFLEX. Ele é fabricado em vários tipos de bitolas e blindagens.
10 - O cabo de cima é AUDIOFLEX. Ele é fabricado com vários tipos de condutores internos.

Agora, se você está pensando que descobriu as diferenças, você errou, porque o de baixo também é KMP; e a KMP tem a mais alta tecnologia em cabos especiais.

AUDIOFLEX®

kmP

Cabos Especiais e Sistemas Ltda.

BR 116/km 25 - Cx. Postal 146 - 06800
Embú SP - Tel. 011/494-2433 Pabx - Telex 011/33234 KMPL - BR - Telegramas Pirelcable

Qualquer dúvida cheme nosso Departamento de Engenharia de Marketing

Trocam-se 8 micros comuns por um Micrão Cobra 480.

Caio

Tratar aqui.

Até hoje, acontecia o seguinte: as empresas pequenas compravam microcomputadores. As empresas grandes compravam computadores grandes. E as empresas que eram grandes demais para um micro ou pequenas demais para um computador grande, compravam um problema.

Agora, você pode trocar o problema por uma solução: o Micrão Cobra 480.

Aliando o desempenho do processamento em 16 bits à possibilidade de ser usado por até 8 pessoas ao mesmo tempo, o Micrão Cobra 480 é uma solução perfeita para quem precisa mais que um micro, mas não quer pagar o alto preço de uma máquina de grande porte.

Com 8 terminais e 5 linhas de comunicação síncrona, capacidade de memória de até 1Megabyte, até 4 unidades de disco rígido Winchester de 10 Megabytes, até 2 unidades de fita de 800/1600, 45 ips, e com impressoras de linha ou seriais, o Micrão Cobra 480 tanto pode resolver todo o processamento de dados de uma empresa de porte médio, como dar conta do processamento distribuído em grandes empresas.

E com uma vantagem que nenhum grupo de micros oferece: o Micrão 480 pode crescer. Como ele é compatível com os computadores de maior porte da Cobra, amanhã sua empresa pode migrar para uma máquina maior, preservando todo o investimento que foi feito em software e periféricos.

Vá até a filial Cobra mais próxima e conheça o Micrão Cobra 480 em detalhes.

Você vai descobrir um computador com as medidas certas em tudo. Até no preço.

MICRÃO Cobra 480

cobra Computadores e Sistemas Brasileiros S.A.

Rio (021) 265-7552
São Paulo (011) 826-8555
Brasília (061) 273-1060 274-9820
Porto Alegre (0512) 32-7111
Curitiba (041) 234-0295
Florianópolis (048) 222-0588
Belo Horizonte (031) 225-4955
Recife (081) 222-0311
Salvador (071) 241-5355
Fortaleza (085) 224-3255

```
1050 XX=XX+1:IFXX<125THEN1055ELSEXX=124
1055 YX=YX+1:IFYX<41THEN106DELSEYX=40
1060 SET(XX,YX):NEXTIX:NEXTKX
1065 FORKX=1TO15:GOSUB1285:NX=INT(10*RND(0)+1)
1070 FORIX=1TONX
1075 XX=XX+1:IFXX<125THEN1080ELSEXX=124
1080 YX=YX-1:IFYX>2THEN1085ELSEYX=3
1085 SET(XX,YX):NEXTIX:NEXTKX
1090 FORKX=1TO25
1095 GOSUB1285:NX=INT(24*RND(0)+1):NX=NX+XX:IFNX<126THEN1100ELSE
NX=125
1100 FORIX=XXTONX
1105 JX=IX+1:AX=POINT(JX,YX):IFAX=-1THEN1130
1110 IFIX()XXTHEN1115ELSEJX=IX-1:AX=POINT(JX,YX):IFAX=-1THEN1130
1115 JX=YX-1:AX=POINT(IX,JX):IFAX=-1THEN1130
1120 JX=YX+1:AX=POINT(IX,JX):IFAX=-1THEN1130
1125 SET(IX,YX)
1130 NEXTIX:NEXTKX
1135 FORKX=1TO35
1140 GOSUB1285:NX=INT(15*RND(0)+1):NX=NX+YX:IFNX<42THEN1145ELSEN
X=41
1145 DX=0:FORIX=YXTONX:AX=POINT(XX,IX):IFAX=0THEN1150ELSEDX=0:GO
TO1175
1150 DX=DX+1:JX=IX+1:AX=POINT(XX,JX):IFAX=-1THEN1175
1155 IFDX()1THEN1160ELSEJX=IX-1:AX=POINT(XX,JX):IFAX=-1THEN1175
1160 JX=XX-1:AX=POINT(JX,IX):IFAX=-1THEN1175
1165 JX=XX+1:AX=POINT(JX,IX):IFAX=-1THEN1175
1170 SET(XX,IX)
1175 NEXTIX:NEXTKX
1180 FORJX=1TO10:GOSUB1305:SX=20:GOSUB1290:GOSUB1310:SX=40:GOSUB
1290:NEXTJX
1185 FORKX=1TO45:IX=INT(121*RND(0)+3):JX=INT(37*RND(0)+3):SET(IX
,JX):GOSUB1320:NEXTKX
1190 PRINT@962,"Modulo de Servico";:PRINT@986,"( QUASAR IV )";:P
RINT@1009,"Energy:";:PRINT@1016,EX;
1195 TX=20:FORIX=1TO20:SX=21-IX:GOSUB1290:NEXTIX
1200 XX=22:YX=31:IX=124:JX=3:SET(XX,YX):SET(IX,JX):TX=10
1205 HX=0:VX=0:KX=0:LX=0
1210 IFXX>=IXTHEN1215ELSEHX=1
1215 IFXX<=IXTHEN1220ELSEHX=-1
1220 IFYX>=JXTHEN1225ELSEVX=1
1225 IFYX<=JXTHEN1230ELSEVX=-1
1230 AX=PEEK(14368):IFAX()16THEN1235ELSEKX=-1
1235 AX=PEEK(14368):IFAX()64THEN1240ELSEKX=1
1240 AX=PEEK(14344):IFAX()4THEN1245ELSELX=-1
1245 AX=PEEK(14344):IFAX()1THEN1250ELSELX=1
1250 EX=EX-1:PRINT@1016,EX;" ";:SX=50:GOSUB1290:IFEX=0THEN1325
1255 RESET(IX,JX):IX=IX+KX:JX=JX+LX:AX=POINT(IX,JX):IFAX=0THEN12
60ELSEIX=IX-KX:JX=JX-LX
1260 SET(IX,JX):IFJX()34THEN1265ELSEIFIX(21THEN1265ELSEIFIX)23TH
EN1265ELSEGOTO1340
1265 RESET(XX,YX):XX=XX+HX:YX=YX+VX:AX=POINT(XX,YX):IFAX=0THEN12
80
1270 IFXX()IXTHEN1275ELSEIFYX()JXTHEN1275ELSEGOTO1355
1275 YX=YX-VX:AX=POINT(XX,YX):IFAX=0THEN1280ELSEYX=YX+VX:XX=XX-H
X:AX=POINT(XX,YX):IFAX=0THEN1280ELSEYX=YX-VX
1280 SET(XX,YX):GOTO1205
1285 XX=INT(123*RND(0)+2):YX=INT(39*RND(0)+2):RETURN
1290 FORAX=1TOTX:OUT255,2:GOSUB1295:OUT255,1:GOSUB1295:NEXTAX:RE
TURN
1295 FORDX=1TOSX:NEXTDX:RETURN
1300 FORDX=1TO32500:NEXTDX:RETURN
1305 A$="     ":PRINT@649,A$;:PRINT@713,A$;:PRINT@777,A$;:RETURN
1310 FORIX=20TO24:SET(IX,35):NEXTIX
1315 SET(20,34):SET(20,33):SET(21,33):SET(23,33):SET(24,33):SET(
24,34):RETURN
1320 TX=5:SX=5:GOSUB1290:RETURN
1325 CLS:PRINT"Modulo de Servico sem energia nao pode prosseguir
":PRINT
1330 PRINT"Tripulacao sem oxigenio...":PRINT
1335 PRINT"E' dado como perdido...":GOSUB1300:GOTO1365
1340 CLS:PRINT"CONGRATULATIONS!":PRINT
1345 PRINT"Modulo de Servico acaba de chegar ao deposito de":PRI
NT"Cristais de Litium da Federacao nesta galaxia.":PRINT
1350 PRINT"TRANSPORTE: Cristais sendo teletransportados...":GOSU
B1300:EX=3000:GOTO1365
1355 CLS:PRINT"SENSORES: Perdido contato com Modulo de Servico":
PRINT:PRINT"Missil Klingon destruiu Modulo.":PRINT
1360 PRINT"Comunicacoes interrompidas...":PRINT:GOTO1335
1365 EX=1#:I=END
10000 END
```

QUASAR/MIX, O JOGO

Aos que suportaram até esta parte, não se desesperem: a listagem 4 é a última e contém o programa principal do jogo. Os 24 Kb de BASIC são o antepenúltimo obstáculo entre o comandante da Enterprise e a conquista do espaço (o último são os Klingons...).

O penúltimo obstáculo são as linhas 1, 6001 e 6002 da listagem 4, pois é preciso inserir nestas linhas três pequenas rotinas em linguagem de máquina, de preferência usando-se o utilitário Pokodes (MS nº 36) embora seja possível tentar fazê-lo de outra forma (com o Superzap, **POKE**s ou **CHR$** ( )...), com o Pokodes é bem mais fácil (veja o Apêndice A, no final do texto).

A linha 1 é um **REM**, com 72 códigos de máquina que devem ser colocados no lugar dos números da linha 1 que servi-

ram para reservar o espaço de 72 bytes (o **REM** fica). É importante que esta linha seja a primeira do BASIC, pois como tal é endereçada por **POKE(16548) + 256 * POKE(16549) + 5** na linha 45. Observe que o +5 é para endereçar a rotina Z80 após os quatro bytes que inicializam cada linha BASIC mais o código (token) de **REM**.

Os 72 códigos de máquina para a linha 1 são: 205, 127, 10, 203, 124, 40, 4, 34, 28, 65, 201, 34, 30, 65, 219, 255, 31, 31, 31, 47, 230, 248, 95,58, 57, 65, 254, 4, 32, 2, 171, 95, 58, 32, 65, 87, 237, 75, 28, 65, 43, 124, 181, 40, 6, 221, 227, 221, 227, 24, 12, 42, 30, 65, 122, 7, 7, 87, 230, 3, 179, 211, 255, 3, 120, 177, 32, 228, 123, 211, 255, 201.

Nas linhas 6001 e 6002 deve-se empacotar nas variáveis strings **X$** e **Y$** as rotinas Z80 restantes, mas sem apagar as aspas normais da string ("). Os números foram colocados para reservar o espaço necessário. O endereçamento é feito por **VARPTR( )** nas linhas subseqüentes.

Os 70 códigos de máquina para a linha 6001 são: 33, 253, 177, 243, 62, 1, 8, 62, 35, 61, 190, 32, 2, 251, 201, 126, 35, 86, 94, 29, 14, 10, 6, 225, 21, 32, 14, 87, 62, 120, 190, 40, 6, 8, 238, 3, 211, 255, 8, 122, 86, 29, 32, 15, 95, 62, 120, 190, 40, 6, 8, 238, 3, 211, 255, 8, 123, 94, 29, 16, 219, 13, 32, 214, 61, 32, 209, 35, 24, 193.

Os 54 códigos para a linha 6002 são: 32, 149, 32, 74, 16, 79, 8, 99, 8, 88, 16, 79, 12, 74, 4, 120, 32, 149, 32, 88, 48, 99, 16, 120, 32, 177, 32, 111, 16, 118, 8, 149, 8, 133, 16, 118, 12, 111, 4, 120, 16, 133, 8, 158, 8, 149, 16, 133, 16, 118, 64, 149, 32, 120.

Após a introdução destes códigos nas linhas 1, 6001 e 6002, estas linhas não poderão mais ser editadas, para que não se perca a informação. Um **LIST** nestas linhas mostrará uma sujeira maluca. Mais uma vez aconselhamos que se use o Pokodes/BAS, mas quem não tem o Pokodes, pode usar o Superzap. E quem não tem o Superzap, utilize **CHR$** ( ) para **X$** e **Y$**, e dê **POKE**s na linha 1 (no endereço já explicado) e substitua o **LPRINT** da linha 2645 por **GOSUB6001(RETURN):LPRINT**.

Veja em seguida alguns comentários que merecem destaque:

– Note na linha 9 o **LOAD** dos módulos compilados com seus **FILINAME**s. A – definição de seus Entry-Points **(DEFUSR)** se dá nas linhas 2734 e 5035 e as chamadas logo em seguida.

– Muitas vezes é feito o teste **IF PEEK(16549) > 80 THEN DEFUSR**... para saber se está em BASIC Disco ou só em cassete, isso foi feito para ajudar a quem só tem cassete. Quem tem BASIC Disco pode eliminar algumas coisas sempre que encontrar este **IF** (que está, por exemplo, nas linhas 20, 45, 2734 e 5035).

– Em algumas linhas como 130 e 160 é usado o caráter ( ]) que normalmente não entra pelo teclado, utilize **CHR$(93)**, concatenando assim as strings para ficar a mesma coisa (ou então faça uso do utilitário Pokodes/BAS).

– Não é necessário posicionar o *memory size* antes de rodar o programa, pois a linha 9 faz isso.

– Na definição de strings do painel de comando nas linhas 130 a 160 e 480 a 550, é importante acertar bem o seu tamanho e não pular nenhum espaço em branco.

– As saídas do jogo ocorrem a partir de 2600 e a avaliação é feita em 2630, variável N.

– Não esquecer as conexões de áudio para o som.

O jogo: instruções e dicas

A missão da Enterprise é patrulhar a galáxia até esgotar o seu tempo, medido em centons, ou até capturar ou destruir todos os inimigos da Federação que estão escondidos pelos diversos quadrantes da galáxia. O comandante da nave tem que enfrentar todos os Klingons, os Atlantis, os Romulans e os Darthlans ■ ainda escapar das dificuldades e imprevistos das viagens pelas dobras espaciais.

No nível fácil (6), mais centons são fornecidos para a execução da missão, porém mais inimigos são encontrados, o que pode ser mais perigoso. A probabilidade de imprevistos ocorrerem neste nível é menor e a viagem transcorre com mais calma, o que é melhor para os iniciantes.

A galáxia é dividida em 8x8 quadrantes de 8x8 setores cada. Os sensores de curta distância (short range sensors) mostram no painel principal de controle da Enterprise o quadrante atual que a espaçonave se encontra e os 8x8 setores do quadrante. A Enterprise é representada por um +, as estrelas por *, uma das bases por **O-O** e os inimigos por diversas formas. É importante observar bem os setores para o cálculo da inclinação de tiro (graus) dos mísseis. As coordenadas do setor do quadrante, e do quadrante da galáxia que a espaçonave está, são mostradas na parte superior do painel principal (e em outros painéis também, para localização).

Os sensores de longa distância (long range sensors) exibem, no centro da tela, as informações sobre o quadrante que a Enterprise está e, ao lado, as informações sobre os quadrantes vizinhos. Essa informação vem na forma de três dígitos, **KYZ**, que indicam a presença ■ o número dos seguintes elementos: inimigos, bases e estrelas.

O computador de bordo sempre armazena as informações colhidas pelos sensores de longa distância e sobre o quadrante atual, mostrando-as reunidas para permitir o estudo da estratégia no painel *galaxy records*. Iniciado o jogo, o que há nos quadrantes (menos o atual) da galáxia é desconhecido.

As bases estelares são poucas, mas se o comandante quiser sobreviver é melhor descobrir logo as suas localizações. Há, entre os Planos de Emergência, a possibilidade de se tentar contato com as bases pelo rádio, buscando assim a localização, mas é grande o risco da base ser localizada e destruída pelos inimigos se o comandante não souber lidar com os códigos secretos de comunicação. E sem base... não há como resistir.

É bom ficar de olho na energia disponível e na energia do campo de proteção (shields). Os shields protegem a espaçonave dos phasers inimigos, mas também atraem os seus mísseis. Os tiros de phasers distribuem a energia total do tiro pelas espaçonaves inimigas presentes no atual quadrante, atingindo com maior intensidade as naves mais próximas e que não estão perto de uma estrela, protegidas. Cuidado para não usar mais energia do que há disponível nem esgotar as reservas logo no início da jornada, pois a recarga geralmente é difícil.

As espaçonaves inimigas contidas em um quadrante não ficam paradas, nem são burras: elas se aproximam para atirar e se afastam quando estão em apuros, e muitas vezes se protegem próximo a uma estrela, que absorve os mísseis ou os tiros de phaser da Enterprise. E muito cuidado se os danos são elevados ou os níveis de energia total ou dos shields ficarem baixos, pois eles têm os seus sensores e se aproximam em formação cerrada (quando há mais de um no quadrante), aproveitando qualquer oportunidade para tentar destruir a Enterprise.

Não hesite em fugir se estiver na pior, mas atenção com o quadrante vizinho: pode ser fatal. Um bom comandante precisa sempre saber se localizar ■ estar pronto para uma ação rápida e correta. Um tiro de phaser destrói um inimigo quando a energia com que este foi atingido é superior à sua energia total disponível. Esta energia só pode ser observada pelos sensores da Enterprise quando um phaser atinge o inimigo. Pode-se usar também o equipamento de emergência, mas gasta-se mais energia e é bem complicado.

Os mísseis são atraídos pelo campo de proteção. Isso é válido para os dois lados, mas nem sempre acertam o alvo: nos limites do quadrante ou perto de uma estrela é bastante difícil acertar ou ser acertado. Cada tiro de phaser que atinge a Enterprise diminui a energia de seu campo de proteção da quantidade de energia com que esta foi atingida. Alarmes de emergência não devem ser negligenciados, pois geralmente indicam que o fim está próximo.

Se uma seção é danificada, não se pode usá-la. Para repará-la são necessários alguns centons, mas isso não deve ser feito durante uma batalha, porque os Klingons não vão ficar esperando.. A máxima velocidade que os motores danificados atingem é de 0.2 Warps, sendo que o normal é 9 Warps. A travessia de um quadrante a outro pelas dobras espaciais exige uma velocidade de impulsão mínima proporcional à distância entre a Enterprise e o limite do quadrante na direção desejada. É possível saltar vários quadrantes de uma vez, adicionando-se 1 Warp por quadrante, mas cautela com o consumo de energia.

Imprevistos e situações difíceis podem ocorrer ao se usar as dobras espaciais. Deformações no espaço podem conduzir a Enterprise para fora da galáxia a grandes velocidades, e aí é preciso manobrar com rapidez para evitar que a énergia acabe ■ não sobre energia suficiente para a volta. Campos de minas Klingon e tempestades muito estranhas podem ser encontradas no hiperespaço, e ambas conduzem ao além.

Abaixo do nível 3 é muito difícil sobreviver às minas hiperespaciais. Para conseguir energia, pode-se ir até uma base, procurar cristais de litium na zona neutra com o módulo de serviço, ou então arriscar a vida em órbita de um planeta de anti-matéria. A base é o mais fácil, mas nem sempre estará disponível. Para chegar até o depósito de cristais de litium é preciso superar o míssel cruise Klingon, que é lançado de uma rampa próxima ao depósito assim que o módulo manobrado manualmente entra na zona neutra (o número de módulos de serviço é limitado). E o planeta de anti-matéria... é melhor deixar para usar só em último caso, mas para quem precisar, é aconselhável ficar em órbita apenas o tempo necessário e não forçar a sorte (este planeta e as minas hiperespaciais costumam ser o cemitério do jogo).

Na hora de atravessar o campo de Quasar IV, que é resultante de explosões de estrelas super-novas, lembre-se que os controles manuais são iguais aos do avião: para baixo sobe e para cima desce. Só se consegue a travessia pelo setor central do campo e sempre acompanhando as linhas de força. Cada partícula atômica que atinge a Enterprise é uma tela de proteção a menos.

A Enterprise acopla-se a uma base apenas encostando nesta (atenção: é para *encostar*, uma colisão pode explodir tudo!), e automaticamente será feita a recarga e a manutenção. Ao usar os motores para se movimentar, lembre-se da localização das estrelas!

Os *radares* quadradar e intergalático são emergências para o caso de navegação às cegas, e são geralmente usados durante combates em que foram destruídos os sensores de curta ou longa distância. Nesses casos, antes de fugir ainda é possível sobreviver usando os antiquados radares. Como radar, porém, não há a indicação do que são os objetos detectados, exceto a própria espaçonave.

Para fugir de um quadrante, numa situação de desespero, resta o black-hole ou buraco negro, pois pode-se chegar ao hiperespaço mergulhando em um black-hole, se a espaçonave conseguir ser controlada na travessia do Túnel de Plasma. É indeterminado o quadrante resultante, havendo ainda o perigo adicional de um afastamento excessivo da galáxia.

A vida do inimigo também tem valor: prisioneiros fazem mais pontos na avaliação do que defuntos. Captura-se uma espaçonave inimiga com a redução de sua energia ao mínimo (quase zero) e depois com a aproximação, pois só então poderá haver a rendição.

Por fim, alguns avisos aos futuros comandantes: a auto destruição é, no mínimo, desaconselhável, porque não há como desistir no meio; para descansar, peça a instrução sobre os comandos; cada manobra efetuada consome energia e tempo, por

isso é fundamental planejar; a avaliação máxima é 100, mas quem chegar vivo ao final dos tempos já é um herói, e quem passar dos 50 pontos é um verdadeiro monstro. Experimente.

- *Apêndice A: Pokodes*

O utilitário Pokodes/BAS (publicado em MS nº 36) tem que ser ajustado para sincronizar os comandos **DATA** contidos nele, no caso do programa co-residente também utilizar comandos **DATA**. Isto é resolvido com a inclusão de duas linhas (mas não esqueça de retirar da linha 65010 o **CLEAR** e o **RESTART**):

```
65005 CLEAR:DATASINCRO
65006 READIS:IFIS<>"SINCRO"THEN65006
```

- *Apêndice B: TRSDOS*

O Compiler/BAS e o Pokodes/BAS rodam normalmente em TRSDOS. Apenas a linha 9 do Quasar/MIX, onde estão os comandos que carregam os módulos Z80 escritos para o NEWDOS, é que têm que mudar para: **CMD"L", "QUASI/USR"** e **CMD"L", "QUASII/USR"**. E quem só tem cassete, vai ter que continuar sofrendo, pois comandos como **DEFUSR**, **DEFFN** e o carregamento acima citado terão que ser refeitos.

BIBLIOGRAFIA

- *80 Micro* (out/82)
- *80 Micro* (dez/82)
- *TRS-80 Assembly Language*, HOWE JR.

Lávio Pareschi é engenheiro eletrônico formado pela PUC/RJ e trabalha na área de Desenvolvimento na Datapoint do Brasil.

Listagem 4 - Quasar/MIX

```
1 REM1234567890123456789012345678901234567890123456789
0123456789012
9 CLS:PRINT"QUASAR IV LOAD GAME.":POKE16561,0:POKE16562,212:PRIN
T:CLEAR400:PRINT"MEMORY SIZE SET AT:";256*212:PRINT"Load machine
 language...":CMD"LOAD QUASI/USR":CMD"LOAD QUASII/USR" 'NewDos
10 RANDOM:RESTORE:DEFINTI-M,S,Q,Z:GOSUB1810:GOSUB6001:GOTO100
20 IA!=VARPTR(SS$):IL!=PEEK(IA!+1):IH!=PEEK(IA!+2):IA!=IH!*256+I
L!:IA!=IA!+(IA!)32767)*65536:IFPEEK(16549))80THENDEFUSR=IA!:RETU
RNELSEPOKE16526,IL!:POKE16527,IH!:RETURN
30 SS$=SE$:GOSUB20:IA!=IA!+10:IB!=IA!+2:POKEIA!,29840/MN:POKEIB!
,29840/MX:U=USR(DR*(MX-MN)/2):RETURN
40 SS$=ST$:GOSUB20:FORIJ=1TO2:FORJI=1TO2:U=USR(JI*20):NEXTJI,IJ:
GOSUB70:RETURN
45 S!=256*PEEK(16549)+PEEK(16548)+5:S!=S!+(S!)32767)*65536:POKE1
6672,S:IFPEEK(16549))80THENDEFUSR=S!:RETURNELSESH!=FIX(S!/256):S
L!=S!-256*SH!:POKE16526,SL!:POKE16527,SH!:RETURN
50 FU=SQR((K1(I)-S1)[2+(K2(I)-S2)[2):RETURN
60 MN=160:MX=460:DR=.5:FORI=1TO1+X/100:GOSUB30:NEXTI:RETURN
70 IFIX=1THENRETURN ELSET=T-.2:PRINT@298,;:PRINTUSING"##.#";T;:R
ETURN
80 FORIK=1TO230:NEXTIK:RETURN
90 FORI=1TO230:U=USR(INT(I/2)+256):NEXTI:RETURN
100 DIMD$(8),K$(3),Q$(3),D(8),K1(7),K2(7),K3(7),S(7,7),Q(7,7),KC
(3,3):W$="  RAZOAVEL!!":P1=.1
110 CLS:PRINT@280,"Q U A S A R  IV":PRINT@940,"By Zorro";TAB(57)
"1984";:SS$=SC$:GOSUB20:GOSUB90:GOTO130
120 SS$=ST$:GOSUB20:FORI7=1TO4:PRINT@IJ,STRING$(LEN(E$),32);:U=U
SR(50)+USR(70):PRINT@IJ,E$;:FORI9=1TO10:NEXTI9,I7:RETURN
130 Q$(0)=" . [+] <==B= * ":Q$(1)=" . [+]=-)0-0 * ":Q$(2)=" . [+
]=-]B=B * ":Q$(3)=" . [+]K-=)O( * ":D$(0)="WARP ENGINES ":D$(1)=
"SHORT RANGE SENSORS "
140 J$="=":D$(2)="LONG RANGE SENSORS ":D$(3)="PHASERS ":O$="ONLY
 GOT":EN$="ENTERPRISE"
150 D$(4)="PHOTON MISSILES  ":D$(5)="GALACTIC RECORDS ":G$="ENER
GY"
160 H$="[   .   ]":F$=" LEFT":D$(6)="SHIELDS ":D$(8)="EMERGENCY":
D$(7)="DAMAGE CONTROL "
170 CLS:PRINT@320,"SELECAO DO MELHOR NIVEL ...:":PRINT:PRINT"1 -
 E Maximo. Ninguem sobrevive, Prometeus.":PRINT"2 - Apenas para
super-homens, Hercules.":PRINT"3 - Para predestinados a herois,
Zorro."
180 PRINT"4 - Impossivel, quase. Napoleao.":PRINT"5 - Aos loucos
, tudo e'possivel....Hitler.":PRINT"6 - Para iniciantes, uma rap
ida destruicao. Dom Pedro."
190 PRINT:PRINT:INPUT"SUA ESCOLHA";LV:IFLV<1ORLV>6THEN2800ELSEK5
=300+100*LV:IE=LV:ZL=7-LV:IC=LV+1
200 SS$=SC$:GOSUB20:FORIW=1TO3:FORIK=54TO59:U=USR(226+IK):NEXTIK
:FORIK=69TO64STEP-1:U=USR(226+IK):NEXTIK:FORIK=1TO20:NEXTIK:NEXT
IW
210 CLS:K$(0)="Klingons  ":K$(1)="Romulans  ":K$(2)="Darthlans "
:K$(3)="Atlantis  ":PRINT"U N I V E R S O  em  F O R M A C A O
.. . ."
220 E0=4000:GOSUB560:Q1=X:Q2=Y:X=8:Z=1:X1=.2075:Y1=6.28:P4=.4:X2
=3.28
230 Y2=1.8:C=100:W=10:P0=W:B9=0:K9=B9:T3=.5:T2=.96:E1=500:K4=B9
240 W1=E0:GOSUB850:FORI=0TO7:FORJ=0TO7:U=USR(330+10*(J-I)):K=0:N
=RND(0):IFN)X1THEN270
250 N=N*64:K=(N<Y1)-Z
260 K=K+(N<X2)+(N<Y2)+(N<.28)+(N<.08)+(N<.03)+(N<.01):K9=K9-K:PR
INT@63+RND(958),"*";
270 B=(RND(0))T2):B9=B9-B:Q(I,J)=K*C+B*W-INT(RND(0)*X+Z):NEXTJ,I
280 T=K9+(LV-1)*10+.2:TT=T
290 T$="MISSIL ":IFK9)T THENT=K9
300 K0=K9:IFB9<ZTHENGOSUB560:Q(X,Y)=Q(X,Y)-P0:B9=Z
310 A=0:IFQ1<0ORQ1)7ORQ2<0ORQ2)7THENN=0:S=0:K=0:GOTO330
320 N=ABS(Q(Q1,Q2)):Q(Q1,Q2)=N:S=N-INT(N/P0)*P0:K=INT(N/100)
330 B=INT(N/P0-K*P0):GOSUB560:S1=X:S2=Y
340 GOSUB45:FORI=0TO7:FORJ=0TO7:S(I,J)=Z:NEXTJ,I:S(S1,S2)=2:FORI
=40TO1STEP-1:U=USR(-200):U=USR(I):NEXTI
350 FORI=0TO7:K3(I)=0:X=8:U=USR(-80):U=USR(2):IFI<KGOSUB590:S(X,
Y)=3:K3(I)=RND(0)*1000
360 GOSUB630:NEXT:I=S:IFB)0THENGOSUB590:S(X,Y)=4
370 IFI)0THENGOSUB590:S(X,Y)=5:I=I-Z:GOTO370
380 IFT<0THEN2600ELSEIF100*RND(0)<=8-LVTHENIFRND(10))5THEN10000E
LSE5000
390 GOSUB780:IFA=0ANDRND(0)<P4THENGOSUB640:GOSUB80
400 IFE<0THENPRINTG$+J$;E:GOTO2610
410 I=Z:H=0:GOSUB570:IFD(I))0THENPRINT:IX=1:GOTO910ELSEIX=0
420 SS$=SC$:GOSUB20:U=USR(270)
430 T=T+.01:GOSUB1970:IJ=70:GOSUB1620:IQ=ABS(INT((Q2*8+Q1)/16)):
IFIQ)3THENIQ=0
440 FORI=0TO7:FORJ=0TO7:PRINT@195+I*64+3*J,MID$(Q$(IQ),3*S(I,J)-
2,3);:NEXTJ
450 ONIGOTO490,480,520,530,500,510,540
460 PRINT@222,"CONDITION  ";C$;" ";
470 NEXTI:GOTO940
480 PRINT@350,D$(6)+"  "+H$;:PRINT@361,;:PRINTUSING"####.#";E1;:
GOTO470
490 PRINT@286,"CENTONS   "+H$;:PRINT@298,;:PRINTUSING"##.#";T;:G
OTO470
500 PRINT@542,"STARBASES "+H$;:PRINT@555,;:PRINTUSING"##";B9;:GO
TO470
510 PRINT@606,"PRISIONER "+H$;:PRINT@619,;:PRINTUSING"##";K4;:GO
TO470
520 PRINT@414,G$+"    "+H$;:PRINT@425,;:PRINTUSING"####.#";E;:GO
TO470
530 PRINT@478,"MISSILES  "+H$;:PRINT@491,;:PRINTUSING"##";P;:GOT
O470
540 PRINT@670,K$(IQ)+H$;:PRINT@683,;:PRINTUSING"##";K9;:GOTO470
550  PRINT" "+D$(0)+"SHUT DOWN";CHR$(30):GOSUB80:D(0)=D(0)+T3:RE
TURN
560 X=INT(RND(0)*8):Y=INT(RND(0)*8):RETURN
570 E=E-H:E1=E1-H:IFE<E1THENE1=E
580 RETURN
590 GOSUB560:IFS(X,Y))ZTHEN590
600 RETURN
610 FORI=0TO7:IFK3(I))0THENS(K1(I),K2(I))=Z:GOSUB590:S(X,Y)=3:GO
SUB630
620 NEXT:RETURN
630 K1(I)=X:K2(I)=Y:RETURN
640 IFK<ZORC$="DOCKED"THENRETURN
650 IG=0
660 FORI=0TO7:IFK3(I)<=0ORK)ZANDRND(0)<.15THENNEXT:RETURN
670 IFIG=1THEN700
680 IX=1:IG=1:CLS:PRINT"QUADRANT";Q1+Z;"-";Q2+Z;:PRINT@49,"SECTO
R";S1+Z;"-";S2+Z;:IJ=25:E$="A L A R M":GOSUB120:PRINT:PRINT:PRIN
T"CONDITION ";C$;
690 PRINT"   SENSORS DETECTION:";K;K$(IQ):PRINTEN$+" "+G$+" LEVE
LS:  SHIELDS:";E1;"  TOTAL:";E
700 IFRND(0)<P4THENIY=1:PRINTK$(IQ)+" "+T$;:H=RND(0)*E1:MN=120:M
X=530:DR=1:GOSUB30:GOTO730ELSEIY=0
710 H=K3(I)*P4*RND(0):K3(I)=K3(I)-H:GOSUB50:SS$=SN$:GOSUB20:U=US
R(100):H=H/(FU[P4):GOSUB570
720 E$=EN$+" FROM":N=E1:E3=Z:GOSUB750:IFH)E1/2THENH=150
730 A1=A:A=1:E3=I:IFH)149THENGOSUB900ELSEIFIY=1THENPRINT" PERDID
O"
740 A=A1:I=E3:NEXT:GOSUB80:RETURN
750 PRINTH;"ACERTOU ";E$;" SECTOR";K1(I)+Z;",";K2(I)+Z;
760 IFE1<0ANDE3=ZTHEN2610
770 PRINT"(restam";N;")":RETURN
780 FORI=S1-ZTOS1+Z:FORJ=S2-ZTOS2+Z:IFI<0ORI)7ORJ<0ORJ)7THEN820
790 IFS(I,J)=4ANDC$="DOCKED"THEN850
800 IFK)0ANDS(I,J)=4ANDRND(0))P4THENPRINT@854,"* UNDOCKABLE *":G
OSUB80:GOTO820
810 IFS(I,J)=4THENC$="DOCKED":T=T-T3:E1=0:PRINTTAB(28)C$:W1=E0:G
OTO850
820 NEXTJ,I:IFK)0THENC$=" RED  ":RETURN
830 FORI=0TO6:IFD(I))0ORE<E0*P1THENC$="YELLOW":RETURN
840 NEXT:C$="GREEN ":RETURN
850 E=E0:P=P0
860 IK=1:FORI=0TO6:IFD(I)<=0THEN890
870 D(I)=D(I)-W1:IFD(I)<=0THENIFIK=1THENCLS:IK=0:FORI7=0TO127:PR
INT@128+I7,CHR$(45);:NEXTI7:PRINT@212,"   (REPAIR TIME)   ":PRIN
TELSE880ELSE890
880 PRINTD$(I);" .......OK.":D(I)=0:GOSUB80
890 NEXT:RETURN
900 I=INT(RND(0)*7):D(I)=D(I)+INT((3-RND(0)*2)*P0)/P0
910 PRINT" *** "+D$(I);"DAMAGED ";CHR$(30):MN=170:MX=1320:DR=.3:
FORIK=1TO4:GOSUB30:NEXTIK:IFD(1))0ANDA<>2THENIX=1
920 IFA=2THENRETURN
930 IFA)2ANDRND(0)<P4THENGOSUB640
940 IFK)0ANDA)2THENGOSUB610
950 IFT<0THEN2600
960 IFE1<1000ORE<3000ORE-E1<200THENGOSUB2130
970 PRINT@832,CHR$(31);
980 CT=K5:A$="":IFC$=" RED  "THENMN=130:MX=820:DR=1:GOSUB30:GOSU
B30:GOSUB70ELSEIFC$="YELLOW"THENMN=120:MX=260:DR=1:GOSUB30:GOSUB
30:GOSUB70ELSEGOSUB40
990 PRINT@832,"Comando ?";:A$=INKEY$:IFA$=""THENCT=CT-1:IFCT=0TH
EN1420ELSE990ELSEA=ASC(A$)-48:PRINTA,
1000 IFA)9ORA<0THENIFRND(10))5THEN5000ELSE10000
1010 IX=0:IFA=0THEN2000
1020 ONAGOTO1060,380,1640,1480,1040,1680,1770,1740,2330
1030 GOTO990
1040 IFD(4))0THENPRINT"    "+T$+" TUBES BLOCKED":GOSUB80:GOTO940
1050 N=15:IFP<ZTHENPRINT"    NO "+T$+"E"+F$:GOSUB80:GOTO940
1060 IFA=5THENPRINT"   "+D$(4)+"SYSTEM ON  [J";
1070 IFA=1THENPRINT@855,D$(0);
1080 PRINT@896,CHR$(30);:INPUT"Curso";C:C=1+C/45:IFC<ZORC)9THEN1
420ELSEIFC=9THENC=0
1090 IFA=5THENP=P-Z:MN=120:MX=460:DR=2:GOSUB30:PRINT@911,"TRACK:
 ";:GOTO1220
1100 PRINT@919,;:INPUT"WARP";W:PRINT@896,;:IFW<=0ORW)9THENGOSUB5
50:GOTO1420
1110 IFW).2ANDD(0))0THENI=0:GOSUB910:PRINT@948,"Max=0.2";:GOSUB8
0:PRINT@896,CHR$(30);:GOTO1100
1120 GOSUB640:GOSUB560:W1=INT(RND(0)*40):IFW1)20RC$="DOCKED"THEN
1190
1130 ONW1GOTO1150,1160
1140 Q1=2*X-P:Q2=2*Y-P:GOSUB2140:GOTO310
1150 GOSUB2210:GOSUB900:GOTO1190
1160 FORI=XTO6:IFD(I))0THEN1180
1170 NEXT:FORI=0TOX:IFD(I)<=0THENNEXT:GOTO1190
1180 D(I)=0:PRINT"   SPOCK FIXED ";D$(I);".  ";:GOSUB80
1190 W1=W/P0:IFW1)=P1THENW1=Z
1200 GOSUB860:N=INT(W*8):E=E-T3*N[2:S(S1,S2)=Z
1210 SS$=ST$:GOSUB20:FORIJ=64TO4STEPW-10:U=USR(IJ):NEXTIJ
1220 Y1=S1+T3:X1=S2+T3:IFT<0THEN2600
1230 Y=(C-Z)*.785398:X=COS(Y):Y=-SIN(Y):FORI=ZTON:T=T-.01:Y1=Y1+
Y
1240 X1=X1+X:Y2=INT(Y1):X2=INT(X1):IFX2<0ORX2)7ORY2<0ORY2)7THEN1
440
1250 IFA=5THENPRINTY2+Z;",";X2+Z;" ";:IFD(1))0THENKU=1002:GOTO12
60ELSEI7=PEEK(16417):I9=PEEK(16416):KU=(X2+Z)*3+128+64*(Y2+1):PR
INT@KU," "+CHR$(140)+" ";:POKE16417,I7:POKE16416,I9
1260 IFS(Y2,X2)=ZORA=5ANDRND(0)<.15THENNEXT:GOTO1380
1270 IFA=ZTHENPRINT"BLOCKED BY ";
1280 ONS(Y2,X2)-3GOTO1360,1330
1290 PRINTK$(IQ);:IFA=ZTHENGOSUB900:GOTO1370
1300 FORI=0TO7:IFY2<>K1(I)THEN1320
1310 IFX2=K2(I)THENK3(1)=0
1320 NEXT:K=K-Z:K9=K9-Z:GOTO1390
1330 PRINT"   STAR";:IFA=5THENPRINT" ABSORBED "+T$;:GOSUB80:GOTO
1420
1340 IFW)RND(0)*P0/2THENGOSUB560:GOTO1140
1350 GOSUB550:GOTO1370
1360 PRINT"STARBASE";:IFA=5THENB=2:GOTO1390ELSEIFW)1THEN2610
1370 Y2=INT(Y1-Y):X2=INT(X1-X)
1380 S1=Y2:S2=X2:S(S1,S2)=2:GOSUB610:A=2:GOTO380
1390 I7=PEEK(16417):I9=PEEK(16416):SS$=ST$:GOSUB20:FORIK=1TO2:PR
INT@KU,MID$(Q$(IQ),S(Y2,X2)*3-2,3);:U=USR(0):PRINT@KU,STRING$(3,
191);:U=USR(0):NEXTIK:POKE16417,I7:POKE16416,I9:PRINT" DESTROYED
";:U=USR(70)+USR(50)
1400 IFB=2THENB=0:PRINTW$:B9=B9-7
1410 S(Y2,X2)=Z:Q(Q1,Q2)=K*100+B*P0+S:IFK9<ZTHEN2630
1420 GOSUB640:IFE<0THEN400
1430 GOSUB780:IFA=5THEN380ELSEIFD(1))0THENIX=1:GOTO940ELSEGOTO94
0
1440 IFA=5THENPRINT"FALHOU";:GOSUB80:GOTO1420
1450 T=T-T2:Q1=INT(Q1+W*Y+(S1+T3)/8):Q2=INT(Q2+W*X+(S2+T3)/8)
1460 IFW1<1THENW1=T2:GOSUB860
1470 Q1=Q1-(Q1<0)+(Q1)7):Q2=Q2-(Q2<0)+(Q2)7):GOTO310
1480 I=3:IFD(I))0THEN910
1490 PRINTTAB(22);:INPUT"PHASERS READY: Units to fire";X:IFX<=0T
HEN1420
1500 IFX)(E-E1)THENPRINTO$;E-E1:GOTO1490
1510 CLS:PRINT"*** PHASERS SYSTEM FIRED ...":PRINT:GOSUB60
1520 E=E-X:Y=K:FORI=0TO7:IFK3(I)<=0THEN1600
1530 IFD(1))0THENX=X*RND(0)
1540 GOSUB50:MN=160:MX=330:DR=2:GOSUB30:H=X/(Y*(FU[P4)):K3(I)=K3
(I)-H:E3=0:E$=K$(IQ)+" AT":N=K3(I)
1550 GOSUB750:IFK3(I)<=0THENE$=K$(IQ)+"  DESTROYED!":MN=240:MX=1
250:DR=1:GOSUB30:GOTO1590
1560 IFK)ZORK3(I))E1/1000ORRND(0)<P4THEN1600
1570 K3(I)=0:IFRND(0)<T3THENE$=K$(IQ)+" EXPLODED":MN=120:MX=1300
:DR=2:GOSUB30:GOTO1590
1580 E$=K$(IQ)+" SURRENDERS":MN=150:MX=380:DR=2:GOSUB30:T=T-T3:K
4=K4+7
1590 PRINTE$:K=K-7:K9=K9-Z:S(K1(I),K2(I))=Z:Q(Q1,Q2)=Q(Q1,Q2)-10
0
1600 NEXT:PRINT:IFK9<ZTHEN2620
1610 GOSUB80:IFK)0THEN1420ELSE380
1620 T=T-.01:PRINT@IJ,D$(1);"AT QUADRANT";Q1+Z;",";Q2+Z;:IFI=1TH
ENIX=0
1630 RETURN
1640 I=2:IFD(I))0THEN910ELSEIX=0
1650 CLS:SS$=SC$:GOSUB20:U=USR(260):IJ=64:GOSUB1620:PRINT:PRINT:
PRINT:PRINT:FORI=Q1-ZTOQ1+Z:FORJ=Q2-ZTOQ2+Z:PRINT" ";
1660 IFI<0ORI)7ORJ<0ORJ)7THENPRINT"***";:GOTO1720
1670 Q(I,J)=ABS(Q(I,J)):GOTO1710
1680 I=5:IFD(I))0THEN910
1690 CLS:IJ=64:PRINT:GOSUB1620:SS$=ST$:GOSUB20:FORIK=0TO12:U=USR
(1)+USR(6)+USR(11):NEXTIK:PRINT:PRINT:GOSUB2120:PRINT:FORI=0TO7:
PRINTI+1;" ";:FORJ=0TO7:U=USR((I+1)*(J+1)):PRINT" ";
1700 IFQ(I,J)<0THENPRINT"***";:GOTO1720
1710 E$=STR$(Q(I,J)):E$="00"+MID$(E$,2):PRINTRIGHT$(E$,3);:GOSUB
20
1720 NEXTJ:PRINT:NEXTI
1730 IFA=3THENFORI=0TO7:GOTO450:NEXTI:GOSUB80:GOTO930ELSEGOSUB80
:IX=1:GOTO930
1740 I=7:CLS:IJ=64:GOSUB1620:PRINT:MN=140:MX=940:DR=1:GOSUB30:FO
RI=0TO6:PRINTD$(I);TAB(21);-D(I):NEXT:PRINT"LAST CENTONS";TAB(21
);T:IFK)0THEN930
1750 PRINT"INIMIGOS"+F$;TAB(21);K9:PRINT:INPUT"CENTONS para REPA
ROS";W1:IFW1<0THENW1=0:IFRND(10)<5THEN5000ELSE10000
1760 T=T-W1:GOSUB860:IFK=0THENIX=1:GOTO930ELSE380
1770 I=6:IFD(I))0THEN910
1780 INPUT"    UNITS TO SHIELDS";N:IFN)E-E1THENPRINTO$;E:GOTO178
0
1790 E1=E1+N:IFE1<0THENE1=0
1800 PRINTD$(6)+J$;E1;:GOSUB80:IFK=0THEN380ELSEIX=1:GOTO930
1810 SE$="":ST$="":SC$="":SN$="":CR$="":GR$=""
1820 FORI=1TO30:READJ:SE$=SE$+CHR$(J):NEXTI
1830 DATA205,127,10,62,1,211,255,237,95,230,255,246,0,87,71,16,2
54,62,2,211,255,66,16,254,43,124,181,32,230,201
1840 FORI=1TO50:READJ:ST$=ST$+CHR$(J):NEXTI
1850 DATA205,127,10,125,254,255,40,38,79,46,160,65,58,61,64,238,
2,50,61,64,211,255,16,252,45,125,183,32,238,180,200,68,197,205,2
27,3,193,225,183,192,126,35,229,96,24,218,68,24,247,32
1860 FORI=1TO30:READJ:SC$=SC$+CHR$(J):NEXTI
1870 DATA205,127,10,62,1,14,255,12,237,91,61,64,69,47,230,3,183,
211,255,13,40,4,16,246,24,242,37,32,241,201
1880 FORI=1TO26:READJ:SN$=SN$+CHR$(J):NEXTI
1890 DATA205,127,10,62,1,211,255,237,95,87,71,16,254,62,2,211,25
5,66,16,254,43,124,181,32,234,201
1900 FORI=1TO20:READJ:CR$=CR$+CHR$(J):NEXTI
1910 DATA205,127,10,1,255,63,125,2,11,120,254,59,32,248,121,254,
255,32,243,201
1920 DIMDD$(9):FORIJ=0TO9:READDD$(IJ):NEXTIJ
1930 DATACOMMANDS ,ENGINES  ,S.SENSORS,L.SENSORS,PHASERS  ,MISSI
LES ,GALAXY   ,SHIELDS  ,DAMAGE   ,EMERGENCY
1940 FORI=1TO29:READJ:GR$=GR$+CHR$(J):NEXTI
1950 DATA33,0,60,126,254,32,32,4,54,191,24,10,203,127,40,6,47,23
0,63,198,128,119,35,62,64,188,32,231,201
1960 RETURN
1970 CLS:FORIJ=0TO11:PRINT@IJ*64,STRING$(64,CHR$(191));:NEXTIJ
1980 FORIJ=1TO8:IFD(IJ-1))0THENPRINT@113+64*IJ," * ";DD$(IJ);" "
;:NEXTIJ ELSEPRINT@113+IJ*64,IJ;DD$(IJ);" ";:NEXTIJ:PRINT@689,"
9 ";DD$(9);" ";:PRINT@113," 0 ";DD$(0);
1990 RETURN
2000 CLS:PRINTTAB(18)"** COMMAND CHOICES **"
2010 PRINT"0 HELP: Comandos e Dicas."
2020 PRINT"1 Navegacao tem direcao (0TO360 graus) & WARP velocid
ade."
2030 PRINT"2 Short Range Sensor mostra o conteudo atual do quadr
ante."
2040 PRINT"3 Long Range Sensor detecta a presenca nos quadrantes
 vizinhos    de Inimigos, Bases, Estrelas."
2050 PRINT"4 Phasers, sistema de ataque distribuido."
2060 PRINT"5 Photon Missiles, sistema de ataque dirigido."
2070 PRINT"6 Galactic Records atualiza imagem da galaxia."
2080 PRINT"7 Shields de protecao. (Cuidado: Os campos de proteca
o atraem     os misseis inimigos)."
2090 PRINT"8 Danos e sistema de Reparos."
2100 PRINT"9 Chamadas de EMERGENCIA. (Na pior...)."
2110 SS$=ST$:GOSUB20:PRINT@960,"HIT...";:U=USR(100):IFINKEY$=""T
HEN2110ELSE380
2120 FORIK=4TO32STEP4:PRINTTAB(IK+1);(IK)/4;:NEXTIK:RETURN
2130 PRINT@832,"TAKE CARE!  LOW ";G$;" LEVELS.";CHR$(30);:MN=170
:MX=1320:DR=2:GOSUB30:RETURN
2140 GOSUB2300:PRINT@88," * SPACE WARP * ";:PRINT@192,"D A N G E
 R :  ";EN$;" GETTING OUT OF GALAXY. "
2150 PRINT@320,"NEW GALAXY QUADRANT: ";Q1+Z;",";Q2+Z
2160 PRINT@448,"POTENCIAL ENERGY  DANGER LEVELS";"   ";
2170 J=RND(26)+64:PRINT@576,EN$;" MUST REVERSE ";D$(0)
2180 PRINT@576,"PRESS URGENTLY ";CHR$(J);CHR$(30):GOSUB2320:IFA$
<>CHR$(J)THEN2200
2190 PRINT"WARP REVERSED":SS$=ST$:GOSUB20:FORIK=24TO8STEP-2:U=US
R(IK):NEXTIK:RETURN
2200 Q1=Q1-Z:Q2=Q2-Z:T=T-P1:IFT<=0THEN2600ELSE2150
2210 GOSUB2300:SS$=CR$:GOSUB20:U=USR(45)+USR(42)+USR(46)
2220 PRINT@74,"D A N G E R";TAB(94);" * ION STORM * "
2230 PRINT@192,"ANTI-MATTER GENERATOR OVERLOAD";TAB(40);G$;" LEV
EL DECREASING"
2240 PRINT@320,"TOTAL ";G$;E;"    SHIELDS:";E1
2250 J=RND(26)+64:PRINT@448,EN$;" MUST DOWN GENERATORS:  PRESS "
;CHR$(J)
2260 GOSUB2320:PRINT@448,CHR$(30)
2270 IFA$=CHR$(J)THENPRINT"GENERATORS CONTROLED":RETURN
2280 E=E-50+50*EXP(-T):IFE<=0THEN2610ELSEIFE<E1THENE1=E
2290 SS$=GR$:GOSUB20:FORI=1TO4:U=USR(0):FORIK=1TO10:NEXTIK:U=USR
(0):FORIK=1TO10:NEXTIK:NEXTI:GOTO2240
2300 FORIK=1TO6:SS$=CR$:GOSUB20:U=USR(32):SS$=ST$:GOSUB20:U=USR(
IK*4):SS$=CR$:GOSUB20:U=USR(RND(64)+127):SS$=ST$:GOSUB20:U=USR(I
K*4+7):NEXTIK
2310 RETURN
2320 FORIK=1TO100:A$=INKEY$:IFA$=CHR$(J)THENRETURNELSENEXTIK:RET
URN
2330 CLS:IJ=21:E$="E M E R G E N C Y":GOSUB120
2340 PRINT:PRINT"0 COMMANDS":PRINT"1 INTERSECTOR RADAR":PRINT"2
INTERGALACTIC  RADAR":PRINT"3 SOS BASE COMUNICATION":PRINT"4 ANT
I-MATTER PLANET FOR RECHARGING":PRINT"5 SURRENDER":PRINT"6 AUTO
DESTRUCTION":PRINT"7 BLACK HOLE"
2350 PRINT"8 ENEMIE REPORT":PRINT"9 RECARREGAR CRISTAIS DE LITIU
M. DEPOSITOS=";IC:PRINT"* Sr Spock lembra perdas de Energia e Te
mpo elevadas....":PRINT:CT=K5
2360 CT=CT-1:IFCT=0THEN380ELSEPRINT@896,"<ENTER>?";:A$=INKEY$:IF
A$=""THEN2360ELSEAA=VAL(A$):PRINTAA;:IFAA<0ORAA)9THEN5000ELSESS$
=ST$:GOSUB20
2370 IFAA=0THEN2000ELSEONAAGOTO2380,2430,2440,2450,2480,2490,250
0,2560,2700
2380 CLS:FORIK=0TO95:SET(IK,0):SET(IK,47):NEXTIK:FORIK=0TO47:SET
(0,IK):SET(96,IK):NEXTIK
2390 FORI=0TO7:FORJ=0TO7:I7=I*128+J*6+2:I9=I7+64:U=USR(10):IFS(I
,J)=1THENPRINT@I7,STRING$(2,176);:PRINT@I9,STRING$(2,131);:NEXTJ
,IELSEPRINT@I7-2,STRING$(6,191);:PRINT@I9-2,STRING$(6,191);:NEXT
J,I
2400 I7=S1*128+S2*6+2:I9=I7+64:PRINT@I7,STRING$(2,143);:PRINT@I9
,STRING$(2,188);
2410 PRINT@50,EN$;:PRINT@114,"Q:";Q1+Z;",";Q2+Z;:PRINT@178,"S:";
S1+Z;",";S2+Z;:PRINT@497,"SECTOR RADAR";
2420 PRINT@945,"HIT..";:GOSUB2580:IFIK=9THEN1420ELSET=T-1:E=E-10
0:CLS:IX=1:GOTO950
2430 CLS:PRINT"QUADRADAR";TAB(40)EN$;" Q:";Q1+Z;",";Q2+Z:PRINT:I
NPUT"WHAT QUADRANT TO RESEARCH (Q1,Q2)";L,M:PRINT:PRINT"RADAR SC
AN...";:FORIK=20TO40:U=USR(IK):NEXTIK:PRINTQ(L-1,M-1)
2431 FORI=0TO7:FORJ=0TO2:II=404+3*I+J:IFJ=1THENPRINT@II,CHR$(140
);:PRINT@II+576,CHR$(140);ELSEPRINT@II,CHR$(188);:PRINT@II+576,C
HR$(143);
2432 NEXTJ,I:FORI=0TO7:PRINT@467+I*64,CHR$(183);:PRINT@492+I*64,
CHR$(187);:NEXTI
2433 JJ=Q1*64+Q2*3+471
2435 FORJI=1TO2:FORI=0TO7:FORL=0TO7:IFABS(Q(I,L)))9THENA$="*"ELS
EA$=" "
2436 FORJ=0TO2:II=469+I*64+L*3+J:PRINT@II,CHR$(191);:IFJ=1THENPR
```

```
INT@II-1,A$;ELSEPRINT@II-1," ";
2437 IFII=JJTHENPRINT@II-3,"(+)";
2438 NEXTJ,L:PRINT@II-24,STRING$(24,128);
2439 NEXTI:NEXTJI:E=E-200:T=T-1:GOTO380
2440 CLS:PRINT"Os covardes nao vivem...":PRINT:GOSUB80:PRINT"Um
gay faria coisa melhor...":GOSUB80:CLS:IJ=24:E$="<( GOS BASE )>"
:GOSUB120:PRINT:PRINT"Alerta vermelho comunicado 'a Frota Estela
r"
2442 FORI=0TO3:FORJ=0TO3:KC(I,J)=RND(101)-1:NEXTJ,I:PRINT"Analis
e dos codigos de Transmissao (X de Incerteza)":PRINT"CODE  K
R   D   A"
2444 FORI=0TO3:PRINTI,:FORJ=0TO3:PRINTUSING"####";KC(I,J),:NEXTJ
:PRINT:NEXTI:PRINT"SOLICITAR 'A BASE SUA POSICAO? (Sr Spock lemb
ra que se a
mensagem for interceptada, babau base...(Y/N)?":GOSUB2580
2446 IFIK=9THEN1420ELSEIFA$()"Y"THEN380
2447 INPUT"Codigo de Tx";I:IFI(00RI>4THEN1420ELSEII=KC(I,I0):PRI
NT"Probabilidade de interceptacao = "II:SS$=ST$:GOSUB20:PRINT"T
X:";FORJ=1TO10:PRINTRND(9),:U=USR(10):NEXTJ:PRINT
2448 PRINT"RX:";FORJ=1TO10:PRINTRND(9),:U=USR(20):NEXTJ:PRINT:P
RINT"DECODER:";:GOTO2650
2450 CLS:S3=RND(8)-1:S4=RND(8)-1:IFS(S3,S4)()1THEN2450ELSEPRINT"
NEAREST ANTI-MATTER PLANET AT SECTOR:";S3*Z;",";S4*Z:PRINT"UNDET
ECTABLE BY SRS":PRINT:PRINTENS;" HAS 20% CHANCES TO EXPLODE IN O
RBIT EACH 400 MEGAJOULES OF REFUEL":PRINT
2460 PRINT"(Y/N)?":GOSUB2580:IFIK=9THEN1420ELSEU=USR(50):IFA$()"
Y"THENT=T-.5:E=E-50:GOTO380ELSEIJ=448:E$="ORBIT! ":GOSUB120:PRIN
T"PRESS 'Z' TO STOP REFUEL":S1=S3:S2=64
2470 PRINT@512,G$;" LEVEL";E:IFRND(0)((ZL/20)THEN261DELSEE=E+100
:T=T-.1:IFINKEY$="Z"THEN380ELSEFORIK=1TO20:NEXTIK:GOTO2470
2480 CLS:IFRND(0)(.5THENPRINTK$(10);" Nao acertaram":GOSUB80:T=T
-1:GOTO380ELSEPRINT"OK!";RND(3000);"Capturados. SCORE 0!":STOP
2490 CLS:CT=10:FORI=1TO09STEP-1:PRINTI,:FORIK=1TO30:U=USR(IK+5):
NEXTIK:NEXTI:GOTO2610
2500 CLS:PRINT"MANUAL CONTROL TO ESCAPE THROUGH A BLACK HOLE":P
RINT"(Z,X) AND ((,))":PRINT:PRINT"MINIMAL CHANCES.(Y/N)?":GOSUB2
580:IFIK=9THEN1420ELSEPRINT@426,"TRAVEL TIME!":IFA$()"Y"THENT=T-
1:GOTO380ELSECT=99:VL=46:VM=24
2510 FORI=0TO8:FORIK=22+2*ITO69-2*I:SET(IK,13+I):SET(IK,35-I):ME
XTIK:FORIK=13+ITO35-I:SET(22+2*I,IK):SET(69-2*I,IK):NEXTIK
2520 FORJ=1TO11:V1=RND(3)-2:V2=RND(3)-2:RESET(VL,VM):VL=VL+V1/2:
VM=VM+V2/2:IFPOINT(VL,VM)=-1THEN261DELSESET(VL,VM)
2530 I7=PEEK(14344):I9=PEEK(14368):V2=0:IFI7=1THENV2=1ELSEIFI7=4
THENV2=-1
2540 V1=0:IFI9=64THENV1=1ELSEIFI9=16THENV1=-1
2550 RESET(VL,VM):VL=VL+V1:VM=VM+V2:IFPOINT(VL,VM)=-1THEN261DELS
ESET(VL,VM):U=USR(20):PRINT@439,CT;"  ";:CT=CT-1:NEXTJ,I:E=E-300
:T=T-3:GOTO1140
2560 CLS:PRINTK$(IQ);" SENSORS":PRINT:FORI=0TO7:IFK3(I)(=0THENNE
XTIELSEPRINTI+1,K1(I)+Z;",";K2(I)+Z,G$;K3(I):NEXTI
2570 PRINT:PRINTENS;" SECTOR:";S1+Z;",";S2+Z:E=E-100:T=T-1:PRINT
"HIT..":GOSUB2580:IFIK=9THEN142DELSEGOTO380
2580 CT=K5:IK=0
2590 A$=INKEY$:IFA$=""THENCT=CT-1:IFCT=0THENIX=1:IK=9:RETURNELSE
2590ELSERETURN
2600 CLS:PRINT:SS$=ST$:GOSUB620:FORIK=1TOK/2+10:I=RND(6):J=RND(9)
:U=USR(I)+USR(J)+USR(9):NEXTIK:PRINT"VOCE NAO E' ETERNO. Acabou
 TEMPO.":GOSUB80:GOTO2620
2610 CLS:E=0:PRINT:PRINT:MN=170:MX=1320:DR=.3:FORIK=1TO5:GOSUB30
:NEXTIK:SS$=SN$:GOSUB20:U=USR(2000):PRINTENS;" DESTROYED!  AH,AH
...AH!"
2620 IFK9=00RT(=0THENCLS:GOSUB45:GOSUB10530:CLS
2625 IJ=384:E$="FROTA ESTELAR:":GOSUB120:PRINT:PRINT"Avaliacao d
e comandante da Enterprise:":PRINT"Inimigos vitoriosos:";K9
2630 IFE1(=0ORE(=0THENM=DELSEN=INT((50*(K0-K9)/K0)+(15*K4/K0)+(1
5*T/TT)+(20/LV))
2640 PRINT:PRINT"YOUR INFINITESIMAL RATING: ";N:IFN)79THENPRINT"
Falha tecnica...":ELSEIFN)59THENPRINTUSELSEIFN(40PRINT"CONDECORAC
AO:  Corte Marcial!":ELSEPRINT"CONDECORACAO: Quem sabe em outra o
portunidade..."
2645 LPRINT:END
2650 FORI=0TO7:U=USR(10):FORJ=0TO7:U=USR(20)
2660 JI!=ABS(@(I,J)/100)-ABS(FIX(@(I,J)/100)):IFJI!)=.1OTHEN2680
ELSENEXTJ,I
2670 PRINT"...INTERFERENCIA INIMIGA IMPEDE RECEPCAO!":GOTO2695
2680 PRINT" X,Y=";(I+1)*(J+1);"  X+Y=";I+J+2:GOSUB80:GOSUB80:T=T
-1:E=E-100:IFRND(1DI)-1)=1ITHEN380
2690 @(I,J)=@(I,J)-10:B9=B9-1:IJ=960:E$="ALERTA VERMELHO":GOSUBi
20:PRINT:PRINT"FROTA ESTELAR INFORMA ... BASE DESTRUIDA!;"
2695 GOSUB80:GOTO380
2700 CLS:IFIC=0THENPRINT"SR. SPOCK: Nao ha'mais depositos de cri
stais de litium
disponiveis na galaxia.":GOSUB80:T=T-1:GOTO380
2705 PRINT"ENERGIA..";E:PRINT"SHIELDS..";E1:PRINT"INIMIGOS:";
K9:PRINT"DEPOSITOS:";IC
2710 PRINT:PRINT"Deseja tentar a travessia da Zona Neutra no Mod
ulo de Servico
para alcancar um deposito de Cristais de Litium (Y/N)?":GOSUB258
0:IFIK=9THEN142DELSEINPUT"MegaJoules p/ o Modulo de Servico";I
2720 IFI)=ETHENPRINT:PRINT"IMPOSSIVEL!!":GOSUB80:T=T-.5:GOTO380
2730 GOSUB45:E$=STRING$(50,32):U=USR(-2000):FORIJ=1TO10:PRINT897
0,E$;:U=USR(100):PRINT8970,"NAVEGACAO: Controles (Z,X) e (<,>).
Atencao...";:NEXTIJ
2734 IFPEEK(16549)>80THENDEFUSR=-3300ELSEPOKE16526,76:POKE16527,
235
2735 E=E-I:U=USR(0):GOSUB80
2740 IC=IC-1:E=E+I:T=T-1:GOTO380
2800 CLS:PRINT"Comecamos mal...":GOSUB80:GOTO2610
5000 CLS:IJ=468:E$="( D A N G E R )":GOSUB120:GOSUB45:E$=STRING$
(64,128):CLS
5005 FORIJ=1TO3:PRINT@0,E$;:GOSUB5100:PRINT@0,"COMPUTADOR: Atenc
ao! Atencao! Foco de particulas gravitacionais.":GOSUB10400:NEXT
IJ
5010 FORIJ=1TO3:PRINT@128,E$;:GOSUB5100:PRINT@128,"SENSORES: Flu
xo gravitacional fortemente concentrado.":GOSUB10400:NEXTIJ
5015 FORIJ=1TO2:PRINT@256,E$;:GOSUB5100:PRINT@256,"ENGENHARIA: M
otores detratores insuficientes. Telas ativadas!":GOSUB1041O:NEX
TIJ
5020 FORIJ=1TO2:PRINT@384,E$;:GOSUB5100:PRINT@384,"SR SPOCK: Alg
lomerado Globular Neutronico. Centro computado.":GOSUB10410:NEXT
IJ
5025 U=USR(-2000):FORIJ=1TO20:PRINT@512,E$;:U=USR(100):PRINT@512
,"NAVEGACAO: Controles (Z,X) e (<,>). Penetrando no campo de..."
:U=USR(300):NEXTIJ
5030 CLS:PRINT@464,CHR$(23);"<<(QUASAR IV)>>":GOSUB10360
5035 IFPEEK(16549)>80THENDEFUSR=-10900ELSEPOKE16526,108:POKE1652
7,213
5040 CLS:U=USR(0):IFI=0THEN2610
5045 CLS:PRINT9968,"SENSORES: Indicam distorcao espacial ultrapa
ssada...";:E1=E1-100:GOSUB80:A$=INKEY$:GOTO380
5100 FORI7=1TO30:NEXTI7:RETURN
6001 X$="12345678901234567890123456789012345678901234567890123456
789012345678901"
6002 Y$="12345678901234567890123456789012345678901234567890123456
789012345678901234"
6003 I1=PEEK(VARPTR(X$)+1):I2=PEEK(VARPTR(X$)+2):I!=I1+I2*256
6004 I!=I!+(I!)32767)*65536
6005 POKE16422,I1:POKE16423,I2
6006 J1=PEEK(VARPTR(Y$)+1):J2=PEEK(VARPTR(Y$)+2):J!=J1+J2*256
6007 POKEI!+1,J1:POKEI!+2,J2:RETURN
10000 GOSUB45:CLS
10030 GOSUB10400
10040 PRINT"A L E R T A   V E R M E L H O!  CURSO DE COLISAO! AT
ENCAO!":GOSUB10400:GOSUB10400:GOSUB10400
10050 PRINT"SENSORES: INDICAM PRESENCA DE MINAS ESPACIAIS A FREN
TE!":GOSUB10400:GOSUB10400:GOSUB10400
10060 PRINT"COMPUTADOR: MANOBRAS EVASIVAS INEFICAZES. RAIOS TRAT
ORES!":GOSUB10400:GOSUB10400:GOSUB10400
10070 PRINT"ENGENHARIA: TELAS ATIVADAS. VELOCIDADE RELATIVA A";R
ND(8);"% LUZ.":GOSUB10410
10080 PRINT"SENSORES: DISTANCIA";RND(900);"MGM. MINAS ANTIMATERI
A KLINGON.":GOSUB10410:GOSUB10410
10090 PRINT"ENGENHARIA: FORCA AUXILIAR LIGADA. VELOCIDADE REDUZI
DA.":GOSUB10420
10100 PRINT:PRINT"COMPUTADOR: A T E N C A O : MANOBRAS MANUAIS (
UP) e (DOWN)...":GOSUB10420
10110 GOSUB10350
10160 U=USR(-3000):U=USR(30):IX=15872:IZ=0:CLS:GOSUB10220:GOSUB1
0280:POKEIX+1,32:POKEIX,62:FORI=1TO250:NEXTI:U=USR(30):POKEIX,32
:POKE16672,5
10170 IY=PEEK(14400):IFIY=8THENIX=IX-65:IFIX(15360THENIX=IX+65:G
OTO10200
10180 IFIY=16THENIX=IX+63:IFIX)16317THENIX=IX-63:GOTO10200
10190 IX=IX+1:IFPEEK(IX)()32THEN10210
10200 POKEIX,62:POKEIZ,32:IZ=IX:FORI=1TO2L:NEXTI
10210 IFPEEK(IX)=42THEN10300ELSEIFPEEK(IX)=46THEN10330ELSE10170
10220 ZR=15360
10230 NN=10+5*IE
10270 IR=RND(NN)+2:ZR=ZR+IR:IFZR)16379THENRETURN ELSEPOKEZR,42:G
OTO10270
10280 IW=63:FORI=1TO8:PRINT@IW,STRING$(I,"."):IW=IW+63:NEXTI:IW
=IW+1:FORI=8TO2STEP-1:PRINT@IW,STRING$(I,"."):IW=IW+65:NEXTI
10290 RETURN
10300 ZB=1:POKE16672,6:POKEIX,191:POKEIX+1,140:POKEIX-1,140:POKE
IX-64,176:POKEIX+64,131:U=USR(-190):U=USR(15):U=USR(-2000):U=USR
(90)
10310 POKEIX,166:POKEIX+1,136:POKEIX-1,132:POKEIX-64,164:POKEIX+
64,137:U=USR(-900):U=USR(100):U=USR(-1000):U=USR(300):U=USR(-100
):U=USR(200)
10320 POKEIX,132:POKEIX+1,128:POKEIX-1,128:POKEIX-64,128:POKEIX+
64,128:U=USR(-1000):U=USR(200):POKEIX,128:U=USR(-5000):U=USR(200
):CLS:GOTO2610
10330 ZB=0:CLS:PRINTCHR$(23):FORI=1TO4:PRINT@256,"SUCCESSFUL PAS
SAGE":U=USR(8000):U=USR(50):PRINT@192,"
  ":FORJ=1TO100:NEXTJ,I
10335 PRINT@256,CHR$(30);"Dificuldade aumentando..."
10340 GOTO10390
10350 CLS:PRINTCHR$(23):PRINT@464,"((( IMPACTO )))"
10360 FORI=8TO2STEP-1:U=USR(-200):U=USR(I)*NEXTI:FORI=19TO1STE
P-1:U=USR(-500):U=USR(I):NEXTI
10370 IJ=898:E$="MANOBRAS EVASIVAS EM MANUAL ***":GOSUB120:PRINT
@898,STRING$(50," ");
10375 IJ=910:E$="ABRIR TELA VISUAL":GOSUB120:GOSUB80:GOSUB45
10380 RETURN
10390 GOSUB10430:A$=INKEY$:GOTO380
10400 FORI=1TO3:U=USR(-3000):U=USR(80):FORJ=1TO30:NEXTJ,I:RETURN
10410 FORI=1TO6:U=USR(-2000):U=USR(10):U=USR(-3000):U=USR(80):FO
RJ=1TO30:NEXTJ,I:RETURN
10420 FORI=1TO7:FORJ=1TO20:U=USR(-100):U=USR(J):NEXTJ,I:RETURN
10430 ZR=1
10460 FORI=1TO100:NEXTI
10480 FORI=1TOZL:U=USR(-2000):U=USR(20):U=USR(15):U=USR(20):U=US
R(-5500):U=USR(15):NEXTI:ZL=ZL+1:IE=IE-1:IFIE=0THENIE=1:ZL=6
10490 IFZL=6THENIR=15360:CLS:PRINTCHR$(23):PRINT@448,"Sorte, por
 enquanto.":GOSUB10540:A$=INKEY$:GOTO380
10500 POKE16772,102:U=USR(-10000):U=USR(20):U=USR(13):U=USR(-300
0):FORI=1TO2:U=USR(15):U=USR(-1000):U=USR(16):U=USR(-3000):U=USR
(18):U=USR(-10000):U=USR(10):U=USR(13):U=USR(-3000):NEXTI:U=USR(
-3000):U=USR(15):U=USR(16):U=USR(15):U=USR(-30000)
10510 U=USR(18)
10520 RETURN
10530 C1=22:GOSUB10550:C1=18:GOSUB10550:C1=15:GOSUB10550:C1=12:G
OSUB10570:C1=10:GOSUB10570:U=USR(9):FORI=1TO100:NEXTI:FORK=1TO2:
GOSUB10500:NEXTK:U=USR(-500):CLS
10540 U=USR(-300):FORI=40TO1STEP-1:U=USR(I):NEXTI:RETURN
10550 U=USR(-10000):GOSUB10590:U=USR(C1):GOSUB10590:U=USR(-2700)
:GOSUB10590:U=USR(C1):GOSUB10590:U=USR(C1):GOSUB10590:U=USR(-100
00):GOSUB10590:U=USR(C1):GOSUB10590:U=USR(-2700):GOSUB10590:U=US
R(C1):GOSUB10590:U=USR(C1):GOSUB10590:U=USR(-6000):GOSUB10590
10555 U=USR(C1)
10560 GOSUB10590:U=USR(-8000):GOSUB10590:U=USR(C1):GOSUB10590:U=
USR(C1):GOSUB10590:U=USR(-2700):U=USR(C1):U=U:U=USR(C1):U=U:U=US
R(C1):RETURN
10570 GOSUB10590:U=USR(-22000):GOSUB10590:U=USR(C1):GOSUB10590:U
=USR(-2700):GOSUB10590:U=USR(C1):GOSUB10590:U=USR(C1):GOSUB10590
:U=USR(C1):GOSUB10590:RETURN
10590 PRINT@472,"Em breve: Quasar V.":OUT255,0:IR=15360+RND(1000
):POKEIR,253:IB=15360+(RND(500)*2):POKEIB,254
10600 PRINT@472,STRING$(20," ");:OUT255,15:RETURN
```


INSTRUMENTOS

* Decida sem dúvidas, erros de informação, falhas de estoque ou vacilações nas entregas.

— Completa linha de instrumentos de teste e medição.

— Garantia de até 2 anos.

— Assistência técnica própria permanente.

— Sistema inédito de reposição quando em garantia.

— Atendimento personalizado para todo o Brasil.

INFORMÁTICA

* Ponha-se em dia com o futuro.

— Microcomputadores Prológica.

— Assistência técnica própria.

— Revendedores em todo o território nacional com a melhor assessoria para ampará-lo no momento de decisão, mesmo que você só precise de uma informação mais precisa sobre os equipamentos.

SUPRIMENTOS CPD

* Unimos o útil ao agradável: qualidade/preço.

— Pronta entrega para todo o território nacional.

— Estoque com os mais variados produtos.

—. Fitas impressoras
. Formulários
. Etiquetas
. Disquettes
. Mesas
. Estabilizadores
. Modens
. Pastas para formulários
. etc.

FILCRES

AJUDANDO A DESENVOLVER TECNOLOGIA

VISITE NOSSO SHOW-ROOM OU SOLICITE NOSSO REPRESENTANTE

FILCRES ELETRÔNICA ATACADISTA LTDA.
Rua Aurora, 165/171/179 — São Paulo — SP
PBX: 223-7388
Vendas São Paulo — Tels.: 220-7954/222-3458
Vendas outros Estados — Tels.: 223-7649/221-0147
Telex: 1131298

A memória do equipamento possui alguns truques que apenas com uma análise mais profunda é possível descobrir, como vamos ver neste artigo

Apple: o mapa da ROM

Aldo Felicio Naletto Junior

A partir deste número, em três artigos sucessivos, o leitor ficará mais familiarizado com as rotinas da memória ROM do Apple. No primeiro trabalho há uma introdução mais ou menos teórica. Na próxima edição apresentaremos o mapa das rotinas da ROM e encerraremos o artigo com um mapa geral sobre a distribuição da memória.

As rotinas das ROMs do Apple formam um autêntico labirinto de Creta, como na mitologia grega, onde o usuário, se não agir como Teseu, guiando-se com um novelo de lã para chegar ao minotauro e voltar, pode se perder. Com o trabalho que agora apresentamos, em sua primeira parte, procuraremos oferecer informações para utilização mais proveitosa do computador.

Antes de mais nada é necessário saber algumas *coisinhas* a respeito dos critérios de ocupação de memória do Apple. Basicamente, os 65536 endereços que o 6502 pode acessar são divididos em quatro faixas, na seguinte seqüência: 2 Kb para a memória do sistema, 46 Kb para os programas e variáveis do BASIC, 4 Kb para entrada e saída ■ 12 Kb para as ROMs (ou EPROMs) do interpretador e sistema operacional.

A memória do sistema é ocupada da seguinte maneira: de $00 a $FE ficam as variáveis do sistema; de $FF até $10F, um buffer que é usado pelo BASIC para traduzir valores binários para strings (como na função STR$); de $110 até $1FF, a pilha do sistema e do BASIC (guarda principalmente dados de FORs, endereços de retorno de sub-rotinas e resultados intermediários de expressões); de $200 a $2FF, o buffer do teclado (onde são armazenados os caracteres que digitamos durante as entradas de dados, linhas de programa ou de comando); de $300 a $3FF fica a área de vetores (na verdade, os vetores só ocupam esta área a partir de $3EF — ou $3CF, caso o DOS esteja presente — e o resto fica livre para o usuário) e de $400 até $7FF a memória de vídeo. Esta serve basicamente para armazenar as 24 linhas do vídeo, o que é feito pelo Apple segundo uma seqüência toda esquisita: em $400 começa a primeira linha, a qual é seguida pela nona em $428, pela décima-sétima em $450 e em $478 por oito bytes que são reservados para uso do cartão que ocupar o slot 0 (estes bytes não aparecem no vídeo), em $480 começa a linha 2, que é seguida em $4A8 pela linha 10, em $4D0 pela 18, e em $4F8 pelos oito bytes reservados ao slot 1. Em $500 temos a seqüência de linhas 3, 11, 19 ■ os bytes do slot 2, e em $580 as linhas 4, 12, 20 etc.

Os 4 Kb de entrada/saída na verdade não contêm memória. O que há são circuitos *pendurados* em certos endereços, de forma que o simples acesso a eles modifica certas características do hardware (como modo texto ou gráfico, alta ou baixa resolução etc.). Há também algumas posições em que o sistema lê dados, como sinais do gravador ou códigos de teclas pressionadas e outras que são reservadas para ROMs dos cartões de expansão. Esta área e a memória serão vistas com mais detalhes na tabela Mapa Geral da Memória, em outro artigo.

Os 48 Kb do BASIC são assim distribuídos: os programas começam na posição $800 (na verdade este endereço contém sempre 00 — é um truque do interpretador — e o programa começa mesmo em $801) e são seguidos primeiro pelas variáveis simples ■ depois pelas indexadas. Após as indexadas começa o espaço string que vai até a posição estabelecida por HIMEM (inicialmente acertada pelo sistema no primeiro endereço após a última página de 4 Kb disponível; quando o DOS está presente, HIMEM é colocado logo abaixo dele, reduzindo a memória disponível para cerca de 35 Kb).

O espaço string é ocupado pelas strings propriamente ditas, isto é, pelas cadeias de caracteres que compõem cada uma delas. Na área de variáveis o que fica mesmo é um conjunto de três bytes para cada string (chamada pela Microsoft de string descriptor (descritor de string) e daqui para a frente referido como DESCR), sendo o primeiro a extensão e os dois seguintes o endereço onde ele realmente está.

O espaço string vai sendo ocupado de trás para a frente, o que significa que cada nova string que aparece é colocada antes das mais antigas. Cada vez que uma delas é alterada, o sistema usa um novo local para armazená-la, deixando sem uso o antigo. Dá para perceber que logo a memória estará entupida de strings sem uso, misturadas às ainda válidas. Quando isso acontece o sistema faz um rearranjo de memória (chamado pela Microsoft de "garbage collection", coleta do lixo), jogando para o final dela as strings válidas e deixando o resto novamente livre.

A memória do BASIC não é apenas ocupada pelo programa ■ suas variáveis. Também as páginas gráficas 1 e 2 de alta resolução ■ a 2 de baixa partilham dela. Estes *inquilinos* são bastante incômodos em certas condições (especialmente a página 2 de baixa resolução, que ocupa o mesmo lugar do primeiro Kb do programa), pois o sistema não *sabe* quando eles estão sendo usados e continua a armazenar coisas ali. A página 2 de baixa resolução fica entre $800 e $BFF, a 1 de alta resolução entre $2000 e $3FFF ■ a 2 entre §4000 ■ $5FFF.

Os últimos 12 Kb são ocupados por ROMs ou EPROMs que contêm o programa interpretador ■ o sistema operacional, sendo que o primeiro ocupa 10 Kb e o segundo os 2 Kb restantes. Este último é quase totalmente auto-suficiente, isto é, não há nenhuma chamada ou salto para as rotinas situadas fora dele, com exceção de alguns *jumps* para as posições $E000 e $E003, que devem conter os pontos de entrada *a frio* e *a quente*, respectivamente, da linguagem ou programa residentes. Para quem não está acostumado com estes termos, ponto de entrada ou partida a frio quer dizer inicialização geral do sistema. Todas as condições iniciais são estabelecidas, começando tudo do zero. Partida a quente, por sua vez, é equivalente ao RESET do Apple: não há perda de dados ou condições correntes do sistema.

O BASIC NO APPLE

Você sabia que o BASIC do seu Apple é interpretado? Isso quer dizer que o programa em BASIC não é convertido para a linguagem de máquina, mas sim fica na memória mais ou menos na mesma forma em que foi digitado, sendo interpretado por um programa monitor, o qual vai reconhecendo as instruções e chamando as rotinas em linguagem de máquina que realmente as executarão.

O programa interpretador consiste, basicamente, em um loop no qual o computador espera que a entrada de uma linha pelo teclado (ou periférico selecionado por IN#), converte-a para um formato comprimido (eliminando espaços e substituíndo as palavras-chaves por códigos de um só byte, chamados *tokens)* e a armazena na memória de programas ou salta para sua interpretação, dependendo de ela ser começada por um número ou não. Em qualquer dos dois casos o sistema sempre acaba retornando ao ponto inicial, onde aguardará a entrada de uma nova linha. Este ponto inicial é conhecido por READY no TRS 80, e será chamado assim também aqui.

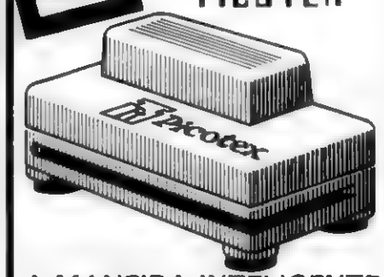
As linhas convertidas e armazenadas na memória ocupam sempre cinco bytes a mais que sua própria extensão. Os dois primeiros são ponteiros que indicam o início da próxima linha, os dois seguintes contém o número da linha atual e o último byte da linha é sempre um 00. A linha mesmo começa no quinto byte e vai até o penúltimo. Então pode aparecer uma dúvida: se os dois primeiros apontam para o início da próxima, como é que fica a última linha do programa, que não tem para quem apontar? Na verdade, este é ■ truque usado pelo sistema para saber quando o programa acabou. A última aponta para uma pretensa linha de apenas dois bytes, ambos 00, isto é, uma falsa linha cujo ponteiro é inválido, já que não há linha apontada armazenada antes de $800.

Na interpretação de uma instrução qualquer, o sistema deve estar sempre inicialmente *olhando* para um byte 00 ou $3A (caráter ":"), caso contrário haverá erro GRAFIA. As posições $B8 e $B9 contêm o endereço para o qual o interpretador está *olhando* a cada instante. Elas constituem uma das mais importantes variáveis do sistema, a qual será chamada aqui de PTRLIN.

O sistema pega os caracteres da linha sob interpretação através de duas rotinas também muito importantes, que são PROXCAR ■ PEGCAR. Estas duas rotinas colocam o caráter apontado por PTRLIN no acumulador e voltam com o Carry resetado se este caráter for um dígito ou com flag Zero setado, caso seja um byte 00 ou um ":". A rotina PROXCAR primeiro incrementa PTRLIN e depois pega o caráter. Já o PEGCAR é, na verdade, uma segunda entrada de PROXCAR, logo após o incremento de PTRLIN, e apenas põe no acumulador o caráter endereçado por ele. Ambas as rotinas ignoram es-

** Nota do autor: Você conhece a lenda de Teseu e o Minotauro? Bem, Teseu foi encarregado de penetrar no intrincado labirinto de Creta e matar o Minotauro, um monstro de cabeça de touro e corpo de homem que morava lá. Teseu não tinha o mapa do labirinto e por isso levou um novelo de lã que foi desenrolando pelo caminho. Assim, Teseu matou o Minotauro, seguiu o fio até a saída do labirinto e entrou para a história.*

ISB DADOS
sistemas de banco de dados
CURSOS DISPONÍVEIS
• Introdução à Microcomputação
• DOS – PC "Sistema Operacional"
• UNIX "Sistema Operacional"
• LINGUAGEM C "Ling. Programação"
• dBASE II "Programação Básica"
• dBASE II "Program. Avançada"
• dBASE III "Program. Básica"
• LOTUS 1-2-3 "Plan. Eletrônica"
• Framework "Sistema Integrado"
• Symphony "Sistema Integrado"
• Wordstar "Processador de Texto"
REG. 561 N.º 0819
MATERIAIS DIDÁTICOS: Publicações Técnicas desenvolvidas em português.
RECURSOS DIDÁTICOS: Conceitos e exemplos práticos, através de Micros e Telão de 72"
CURSOS FECHADOS E ABERTOS
CONTATOS PELO TEL: (011) 285-0132 - Al. Santos, 336 - cj 42 CEP 01418 - SP

chegou PICOTEX

A MANEIRA INTELIGENTE DE FAZER ECONOMIA
Duplica a capacidade dos seus disquetes 5¼

Faça já o seu pedido
Preço de Lançamento:
Cr$ 49.000,00
cervadin
Rua José Maria Lisboa, 580
Tel.: 287-4716 - CEP 01423
Jd. Paulista - São Paulo - SP

paços em branco, saltando por cima deles até encontrarem o caráter válido. Estas rotinas estão originalmente gravadas na ROM, mas são transferidas para o início da RAM durante a inicialização do BASIC. A rotina PROXCAR começa em $00B1 e a PEGCAR em $00B7.

Como já foi visto antes, as variáveis simples começam logo após os dois bytes 00 do fim do programa, e são seguidas pelas variáveis indexadas e pelo espaço string. Os endereços de início do programa, das variáveis simples, das indexadas e do espaço string são guardados nas variáveis dos sistemas INIPROG, INIVARS, INIMATR e INISTR.

As variáveis simples ocupam sempre sete bytes, sendo dois para o nome e cinco para o valor. Uma variável tem os bits 7 dos dois bytes do nome setados e usa apenas dois dos cinco bytes restantes para o valor. Uma variável real tem os bits 7 do nome zerados e ocupa todos os cinco bytes, sendo o primeiro o expoente e os quatro seguintes a mantissa (mais significativo primeiro). As variáveis string têm o bit 7 do primeiro byte do nome zerado e o do segundo setado, e usam apenas três bytes para o valor, que na verdade é o descritor de string mencionado anteriormente.

As variáveis indexadas obedecem às mesmas regras para nome, porém, ocupam dois, três ou cinco bytes por elemento, dependendo do tipo. Além destes, cada variável ocupa mais 5+2* N bytes (onde N é o número de dimensões), sendo os dois primeiros usados para o nome, os dois seguintes para o total de bytes gastos e um byte para o número total de dimensões e mais dois para cada dimensão, que indicam qual o valor máximo de cada uma (primeiro a dimensão, que aparece por último no índice).

ACUMULADORES

O conceito de acumulador é muito importante para entender a resolução de expressões. O acumulador é uma variável do sistema que sempre contém um dos operandos e na qual também ficará o resultado da operação. Há três acumuladores no BASIC, um para cada tipo de valor: o ACSINT ocupa as posições $A0 e $A1, sendo usado para valores inteiros; o ACSTR1 fica também em $A0 e $A1 e retém o PTRDESCR (que é o endereço em que está o DESCR) da string-resposta (nas funções e expressões string, o DESCR do resultado fica nas posições $9D, $9E e $9F) e o ACSPF1, que recebe os valores reais (em ponto flutuante), ocupa as posições $9E até $A1 com a mantissa (mais significativo $9E) e $9D com o expoente. O termo ACSOFT1 servirá de nome genérico para os três acumuladores. O tipo do valor a ser armazenado lá é o que dirá qual dos três será usado. A variável do sistema TIPOAC, que ocupa as posições $11 e $12, informa o tipo do valor contido no ACSOFT1. A posição $11 indica valor numérico se for 00 e string se for 255, enquanto o desempate entre valores numéricos inteiro e em ponto flutuante é feito pela posição $12 — 00 para ponto flutuante e 128 para inteiro.

Valores diferentes em TIPOAC não estão definidos e confundirão o computador, provocando muitos erros tipo DIFERE. Algumas funções internas (+,–,*, /, ^ ,AND e OR) requerem dois operandos. Nestes casos são usadas variáveis auxiliares para conter o primeiro operando, ficando o segundo no ACSOFT1. Por analogia, estas variáveis auxiliares serão chamadas ACSOFT2 (nome genérico), ACSTR2 ($A8/$A9) e ACSPF2 ($A5/$A9). Não existe ACSINT2 porque toda a matemática do Apple é em ponto flutuante, sendo usado o ACSINT apenas nas conversões inteiro/ponto flutuante e ponto flutuante/inteiro.

BIBLIOGRAFIA

No levantamento das rotinas da ROM foram usados apenas três livros: o "6502 Software Design", de Leo J. Scanlon (Serie Blacksburg/Howard Sams & Co, Inc.), o "Guia de Usuários do Apple II", de Lon Poole, Martin McNiff, Steven Cook (Osborne/McGraw Hill) e o "Apple II Circuit Description", de Winston D. Gayler (Howard Sams & Co, Inc.).

Os dois últimos são bastante recomendáveis. O primeiro é como o manual do Apple deveria ser, e o segundo dá explicações detalhadas de como o circuito do Apple funciona, além de esquemas, diagramas de tempo, etc. O livro traz ainda uma tabela de rotinas mais ou menos como a deste artigo, porém ela fica restrita à ROM do sistema operacional, além de ser menos completa; por outro lado, ela diz quais registradores são alterados em cada rotina, o que às vezes é muito útil.

O motivo da bibliografia ser tão pequena é que este artigo não é nenhuma *tradução* de livro americano: ele é resultado de mais de um ano de *xeretices* em cima de listagens disassembladas do Unitron e do TRS 80 Dismac D8000. Paradoxalmente, trabalhar nas duas máquinas ao mesmo tempo facilita as coisas; isto ocorre porque, como os BASIC dos dois são versões reduzidas do MBASIC da Microsoft, pode-se extrair a estrutura básica do sistema através da comparação das rotinas, semelhantes nos dois computadores.

Os nomes de rotinas ou variáveis são quase todos criações minhas, não tendo nada de oficiais. O artigo está longe de dar uma cobertura completa ao assunto, o que exigiria um livro. Mas fornece uma boa base para que os leitores pesquisem mais a fundo as rotinas de seu interesse. Mais para a frente pretendo publicar o mapa da ROM do TRS 80 e artigos específicos sobre o interpretador, suas rotinas matemáticas e de manipulação de strings, assuntos bastante complexos.

Aldo Felicio Naletto Junior tem 26 anos, é engenheiro eletrônico pela Escola de Engenharia de São Carlos, da USP, trabalha no Projeto CATE da Telebrás no Laboratório de Eletretos do Instituto de Física e Química de São Carlos e na agência do Banco do Brasil em São Carlos. Mantém com um sócio uma empresa de processamento de dados e implantação de sistemas.

Mensagem de erro

Em MS n.º 41, na matéria A Lógica na Programação, quinto parágrafo, sétima linha, apareceu, indevidamente a palavra indiferença, o correto é: ... As principais operações são união, interseção, diferença...
Na figura 5, foram publicadas as linhas A 3 OR B 7 e CS = "FALSO" AND NOT B 3. Nas duas linhas, faltaram os sinais > e < . O certo é: A > 3 OR B > 7 e CS = "FALSO" AND NOT B < 3.

Na figura 8, as operações união, interseção, diferença, produto cartesiano, projeção, restrição e junção foram impressas sem setas. Assim, a forma correta seria, por exemplo, T ← A U B, e não T AUB. Faltou, ainda, na operação interseção, o sinal ∩ .A forma certa é T ← A ∩ B.

OFERTA DO MÊS

Mesas para terminais de vídeo
Cr$ 419.850
• Fabricação própria
• Cores discretas
• Desenho moderno
• 5 modelos

Na compra de Cr$ 1.000.000 você ganha um aparelho que duplica a utilização do diskete

COMPUTADORES
• Suprimentos
• Periféricos
• Impressoras
• Drives
• Placas de Expansão Interfaces
• Cabos

Conosco você encontra também, tudo o mais que precisa em vídeo-game, som, telefonia, das melhores marcas e procedências, e mais:

VÍDEOS
• Transcodificação todos os sistemas
• Fitas: VHS - BETA-U-MATIC e para limpeza de cabeça
• Baterias p/2 e 8 hs.
• Iluminadores
• Cabos de extensão p/câmeras
• Bolsas p/câmeras e vídeos
• Telão

• Acessórios nacionais e importados
• Suporte p/ TV teto ou parede
• Curso de inglês em vídeo-cassete
• Serviço expresso remetemos para todo Brasil

BTC" 2001 ALTA TECNOLOGIA

BRASIL TRADE CENTER

Av. Epitácio Pessoa, 280 (Esq. de Visconde de Pirajá), Ipanema - Rio de Janeiro - CEP 22471 - 259-1299
Rua da Assembléia, 10 - Loja 112 (Ed. Cândido Mendes) Rio de Janeiro - (021) 222-5343
Av. das Américas, 4790 - Sala 615 (Centro Profissional Barra Shopping) Rio de Janeiro - 325-0481
TELEX (021) 30212 BTCP
Fábrica: Rua Silva Vale, 416 - Cavalcanti - RJ - Tel.: (021) 592-3047

Nesta segunda parte do artigo a orientação para utilização dos comandos complementares dos arquivos do NEWDOS/80

Arquivos em disco do NEWDOS/80

João Henrique Volpini Mattos

Complementando artigo cuja primeira parte foi publicada em MS 39, vamos agora praticar os novos comandos utilizados com os arquivos do NEWDOS/80. Antes de continuar, é aconselhável uma releitura da parte inicial, pois são muitos os termos técnicos utilizados pelo NEWDOS/80.

É necessário muito cuidado na digitação das instruções em todos os exemplos a seguir. A execução de uma instrução errada poderá prejudicar toda uma seqüência de exemplos. Se isso ocorrer, retorne ao programa utilizado para criar o arquivo e execute todos os exemplos, novamente.

ARQUIVOS MU

Este tipo de arquivo foi inicialmente concebido para substituir os arquivos seqüenciais do TRSDOS (**PRINT/INPUT** no NEWDOS/80), oferecendo algumas vantagens: grava os valores numéricos na sua representação binária (e não em ASCII), permite a alteração de registros (obedecendo certas restrições) e possibilita o acesso randômico através de índices. Uma característica deste arquivo é o fato dele marcar o início de cada registro e de cada campo com determinados bytes identificadores:

- **70H** – Indica o início de um registro (SOR-Start Of Record). Todo o início de um registro é marcado com o byte 70H, mas nem todo o byte 70H indica o início de um registro, pois ele pode aparecer como parte de valores numéricos ou em strings (letra p minúscula).
- **72H** – Indica que os dois bytes a seguir são um valor inteiro.
- **73H** – Indica que os quatro bytes a seguir são um valor real de precisão simples.
- **74H** – Indica que os oito bytes a seguir são um número real de precisão dupla.

As strings são identificadas de dois modos: se ela tem menos de 128 caracteres, o byte indicador do campo será o resultado da soma de **80H** mais o número de caracteres da string. Se ela tem 128 ou mais caracteres, a marcação será feita por dois bytes: um **71H** e outro indicando o comprimento da string.

Para familiarização com este tipo de arquivo, nada melhor do que trabalhar um pouco com ele. De início, um pequeno programa com a listagem 1, a seguir. É importante não esquecer o ponto e vírgula no fim dos IGEL (Item Group Expression List) nas linhas 3, 4, 5 e 7.

```
1 CLEAR 1000
2 OPEN "O",1,"EXEMPLO/MU","MU"
3 PUT#1,"RIO DE JANEIRO";
4 PUT#1,STRING$(130,"*");
5 PUT#1,"NITEROI","CABO FRIO";
6 V0 = "30" : V1 = 31 : V! = 32.0001 : V# = 33.00000000001
7 PUT#1,V0,V1,V!,V#
8 CLOSE
```

Execute o programa com **RUN**, criando o arquivo Exemplo/MU. Volte ao sistema operacional com **CMD"S"** e chame o **SUPERZAP**. Com a opção do DFS (Display File Sector) analise o setor 0 do arquivo Exemplo/MU (Figura 1). Para facilitar a identificação dos registros e campos foi feito um círculo em torno dos bytes SOR dos registros e sublinhados os marcadores dos campos.

No início do setor vemos um byte **70H** (SOR) e logo a seguir um **8EH**, indicando que a seguir vem uma string de 14 bytes (RIO DE JANEIRO), pois a diferença **8EH–80H = 0EH**, que é 14 em decimal. Após a string, temos novamente um SOR e a seqüência de bytes **71H** e **82H**(**82H** =130 decimal), indicando que a seguir temos uma string de 130 caracteres (veja linha 4 da listagem 1). Após os 130 asteriscos, lá na posição **95H**, temos um novo SOR, apontando o início do registro em que foram gravadas as strings NITEROI e CABO FRIO. Identifique os bytes marcadores destes campos. Finalmente, no último registro (posição **A8H**, na figura 1), temos os quatro campos gravados na linha 7 do programa. A string 30 é facilmente identificável, mas os valores numéricos 31, 32.0001 e 33.00000000001 não o são, pois eles estão armazenados em sua forma binária:

- inteiro 31 = IF 00, na representação binária;
- real 32.0001 = 1A 00 00 B6, na representação binária;
- duplo 33.00000000001 = FB 2B 00 00 00 00 04 86, na representação binária.

Voltando ao BASIC, escreva agora as linhas da listagem 2, abaixo.


```
1 PRINT "Teste de fim de arquivo          :";LOC(1)=0
2 PRINT "Posicao do EOF                   :";LOC(1)*2
3 PRINT "Posicao do proximo registro      :";LOC(1)+1
4 PRINT "Posicao do ultimo registro acessado :";LOC(1)=0
5 STOP
```

Não dê **RUN** no programa, ao invés disso digite:

```
CLEAR 1000 : OPEN "R",1,"EXEMPLO/MU","MU" : GOTO 1
```

O sistema responderá:

```
Teste de fim de arquivo          : 0
Posicao do EOF                   : 189
Posicao do proximo registro      : 0
Posicao do ultimo registro acessado :
BAD FILE MODE in 4
```

Observe que a função LOC(1) # , que indica a posição do último registro acessado (REMRA – Remembered Record Address) resultou em erro. Isso ocorre porque nenhum registro foi ainda lido ou gravado e o sistema invalida a função. Façamos então a leitura do primeiro registro. Para facilitar a digitação, substituímos o **PRINT** pela interrogação "?" :

```
GET 1,,A$ : ? A$ : GOTO 1
```

A resposta será:

```
RIO DE JANEIRO
Teste de fim de arquivo          : 0
Posicao do EOF                   : 189
Posicao do proximo registro      : 16
Posicao do ultimo registro acessado : 0
```

Note que os valores relacionados ao EOF (End of File) não se alteraram, pois estamos fazendo uma leitura de dados. Apenas as posições do próximo registro e do último acessado foram alteradas. (Na verdade, os ponteiros que indicam estes registros.) Nosso arquivo agora está posicionado no início da string de 130 asteriscos. Vamos ler apenas os 10 primeiros:

```
GET 1,,,10,A$ : ? A$ : GOTO 1
```

E teremos:

```
**********
Teste de fim de arquivo          : 0
Posicao do EOF                   : 189
Posicao do proximo registro      : 149
Posicao do ultimo registro acessado : 16
```

Observe que foi possível a leitura parcial de um campo. Somente o NEWDOS/80 oferece esta flexibilidade. Vejamos agora o próximo registro:

```
GET 1,,A$,B$ : ? A$,B$ : GOTO 1
```

O sistema responde:

```
NITEROI          CABO FRIO
Teste de fim de arquivo          : 0
Posicao do EOF                   : 189
Posicao do proximo registro      : 168
Posicao do ultimo registro acessado : 149
```

Vamos ler este registro novamente, utilizando o FP (File Position), que posiciona o arquivo no início do último registro acessado:

```
GET 1,0,,C$,D$ : ? C$,D$ : GOTO 1
```

Teremos então:

```
NITEROI          CABO FRIO
Teste de fim de arquivo          : 0
Posicao do EOF                   : 189
Posicao do proximo registro      : 168
Posicao do ultimo registro acessado : 149
```

É importante observar que as posições do próximo registro e do último registro acessado não se alteraram. Façamos agora a leitura do segundo e quarto campos do próximo registro. Observe os *nulos* entre as vírgulas, indicando que o campo correspondente deverá ser pulado:

```
GET 1,,,,N%,,N0$ : ? N%,N0$ : GOTO 1
```

O sistema responderá:

```
31               33.00000000001
Teste de fim de arquivo          :-1
Posicao do EOF                   : 189
Posicao do proximo registro      : 189
Posicao do ultimo registro acessado : 168
```

```
DRV 00  70BE 5249 4F20 4445 204A 414E 4549 524F  p.RIO.DE.JANEIRO
0   10  7071 822A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  pq.**********
OH  20  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
    30  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
DRS 40  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
65  50  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
41H 60  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
    70  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
    80  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
    90  2A2A 2A2A 2A70 874E 4954 4552 4F49 8243  ******p.NITEROI.C
    A0  4142 4F20 4652 494F 7082 3330 721F 0073  ABO.FRIOp.30r..s
    B0  1A00 00B6 74FB 2B00 0000 0004 B600 0000  ....t.+.........
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000  ................
0   D0  0000 0000 0000 0000 0000 0000 0000 0000  ................
OH  E0  0000 0000 0000 0000 0000 0000 0000 0000  ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figura 1

É importante observar que acabamos de ler o último registro do arquivo, pois a posição do EOF é igual à posição do próximo registro (189). Outro modo de verificar isso é através do resultado do teste de fim de arquivo, que está indicando –1 (verdadeiro). Vamos então retornar ao início deste registro para ler os seus outros dois campos:

```
GET 1,0,,N$,,N%! : ? N$,N% : GOTO 1
```

A resposta será:

```
30               32.0001
Teste de fim de arquivo          :-1
Posicao do EOF                   : 189
Posicao do proximo registro      : 189
Posicao do ultimo registro acessado : 168
```

O que aconteceria se tentássemos ler mais um registro? Vamos ver:

```
GET 1,,,A$ : ? A$ : GOTO 1

END OF FILE ENCOUNTERED
INPUT PAST END
```

Aconteceu o que era de se esperar.

Vejamos agora outra técnica de posicionamento de arquivo. Pela análise do arquivo com o **SUPERZAP** vimos que o terceiro registro começava na posição **95H** (149 em decimal). Digite então:

```
GET 1,'149,,B$,D#! : ? B$,D# : GOTO 1
```

O sistema responderá:

```
NITEROI          CABO FRIO
Teste de fim de arquivo          : 0
Posicao do EOF                   : 189
Posicao do proximo registro      : 168
Posicao do ultimo registro acessado : 149
```

A utilização de valores determinados de RBA (Relative Byte Address), seja um número, o conteúdo de uma variável ou o resultado de uma expressão, fará com que este valor seja transferido para o ponteiro do próximo registro. Isso nos permite acessar o arquivo de forma randômica, bastando para isso armazenar os RBA dos registros num vetor e acessá-los através dele. Digite o programinha a seguir, listagem 3, e dê um **RUN 100**. Não apague as linhas 1 a 5 que já estão na memória do computador, pois continuaremos a utilizá-las.

```
100 OPEN "R",1,"EXEMPLO/MU","MU"
101 I = 0
102 I = I+1 : GET 1 : RB(I) = LOC(1)% 'pega o RBA do ultimo registro acessado
103 IF NOT LOC(1)% THEN 102 'verifica se ja chegou ao fim do arquivo
104 CLOSE
105 CLS : PRINT "Numero de registros no arquivo :"; I
106 FOR J = 1 TO I : PRINT "Registro"; J; "comeca no RBA"; RB(J) : NEXT
107 END
```

Com o RUN 100, o sistema responderá:

```
Numero de registros no arquivo : 4
Registro 1 comeca no RBA 0
Registro 2 comeca no RBA 16
Registro 3 comeca no RBA 149
Registro 4 comeca no RBA 168
```

Em arquivos *de verdade* não se esqueça de dimensionar o vetor onde serão armazenados os RBA. Neste caso não foi preciso, pois sabíamos que o arquivo tinha menos de 10 registros. Não é necessário dimensionar vetores com menos de 10 elementos.

Vejamos agora alguma coisa de gravação em arquivos MU. Como você já deve ter imaginado, a alteração dos registros existentes ficará condicionada a que o comprimento do novo registro (incluíndo os bytes marcadores), seja igual ou menor

ao registro já existente. Vamos então abrir o arquivo e alterar o primeiro registro:

```
OPEN "R",1,"EXEMPLO/MU","MU"
PUT 1,,,"NOVO RIO"! : GOTO 1
```

O sistema responderá:

```
Teste de fim de arquivo                 : 0
Posicao do EOF                          : 109
Posicao do proximo registro             : 16
Posicao do ultimo registro acessado     : 0
```

Perceba que aconteceu o mesmo quando fizemos um GET no primeiro registro. Vamos agora dobrar o valor armazenado no terceiro campo do quarto registro (o valor real de precisão simples 32.0001), para exemplificar a leitura/gravação parcial de registros. E também aproveitar o vetor RB, que não deve ter sido apagado da memória.

```
GET 1,!RB(4),,,! 'posiciona o arquivo no inicio do terceiro
                  campo do quarto registro.
GET 1,*,,V!!     'le a variavel V! a partir da posicao atual
                  do arquivo.
PUT 1,$,,2*V!!   'grava o dobro de V! a partir do ponto onde
                  foi realizada a ultima transferencia de
                  dados.
GOTO 1
```

Eis a resposta:

```
Teste de fim de arquivo                 : -1
Posicao do EOF                          : 109
Posicao do proximo registro             : 109
Posicao do ultimo registro acessado     : 1??
```

Novamente nos encontramos no final do arquivo. Vamos estendê-lo. Digite:

```
PUT 1,!%,,"ULTIMO REGISTRO"! : GOTO 1
```

A resposta:

```
Teste de fim de arquivo                 :-1
Posicao do EOF                          : 206
Posicao do proximo registro             : 206
Posicao do ultimo registro acessado     : 189
```

Será que nosso arquivo foi relamente estendido? Na realidade ainda não, pois a string Último Registro ainda está no buffer do arquivo, na memória do computador, e só será gravada quando dermos o próximo **PUT** ou **GET** ou fecharmos o arquivo. Para forçar a gravação deste registro podemos utilizar o comando **PUT 1,&** mas o diretório ainda estará desatualizado. Para forçarmos a gravação do registro e a atualização do diretório, antes que alguma queda de luz o prejudique, basta darmos um **PUT 1, &&** e ele estará então atualizado.

Vamos agora ao segundo registro (a string de 130 asteriscos), alterando-o com alguns campos menores:

```
PUT 1,!RB(2),,STRING$(40,"A"),STRING$(40,"B"),STRING$(40,"C")! : GOTO 1
```

Resposta do sistema:

```
Teste de fim de arquivo                 : 0
Posicao do EOF                          : 206
Posicao do proximo registro             : 149
Posicao do ultimo registro acessado     : 16
```

E se tentássemos colocar mais uma outra string de 40 caracteres neste registro? Vamos tentar, mas antes, atualizemos o arquivo:

```
PUT 1,&&  'atualiza o arquivo e o diretorio no disquete
PUT 1,*,,STRING$(40,"D")! : GOTO 1

RECORD OVERFLOW
```

É... estourou o registro. Mas será que pelo menos uma parte da string foi gravada? Feche o arquivo (**CLOSE**), saia do BASIC (**CMD"S"**) e volte a examinar o arquivo com a opção DFS do **SUPERZAP** (figura 2). Observe que além dos bytes SOR nas posições **00H**, **10H**, **95H** e **A8H**, que já existiam anteriormente, temos agora um outro SOR na posição **BDH**, correspondente ao registro que foi adicionado ao arquivo. Veja agora o primeiro registro. O byte marcador (**88H**) indica uma string com 8 bytes de comprimento (**88H–80H = 08H**). Percorrendo 8 bytes, a partir do início da string NOVO RIO encontramos uma seqüência de bytes 00. Estes bytes são utilizados como *enchimento* pelo sistema, já que NOVO RIO tem um comprimento menor que RIO DE JANEIRO, anteriormente gravada.

No registro seguinte vemos as strings de 40 A's, B's e C's e logo após uma nova seqüência de 00, indicando que a string de 40 letras **D** que queríamos gravar não foi transferida para o arquivo nem ao menos parcialmente. No NEWDOS/80 um erro durante a execução de **PUT/GET** faz com que os dados do registro não sejam transferidos e que a posição atual do arquivo não se altere.

No próximo registro temos a string 30, o número inteiro 31 (1F 00 na representação binária), o real 64.0002 (1A 00 00 B7) e a seguir zeros. O valor 33.00000000001 (precisão dupla) simplesmente foi apagado do arquivo. Isso mostra o cuidado que devemos ter na alteração parcial de registros, pois tudo o que vier após o último campo gravado será perdido.

ARQUIVOS TIPO MF

Na primeira parte deste arquivo você conheceu as características dos arquivos MF. Todos os seus campos são marcados do mesmo modo que nos arquivos MU. E todos os registros têm o mesmo comprimento, definido na abertura dos arquivos.

Assim como nos arquivos MU, os registros MF podem ser alterados, não sendo obrigatório manter o mesmo número, tipo ou comprimento dos campos do registro original. A única orientação a obedecer é o comprimento máximo do registro.

Utilizando a mesma técnica dos arquivos MU, vamos executar algumas instruções com arquivos MF. Comecemos digitando o seguinte programa da listagem 4:

```
1 CLEAR 1000
2 OPEN "D",1,"EXEMPLO/MF","MF",[illegible] 'especifica registros de 64 bytes
3 PUT 1,,,"CAMPO1","CAMPO 2","CAMPO NO.3"!
4 PUT 1,,,STRING$([illegible],"*")! 'maior string possivel
5 [illegible]
6 PUT 1,,,I#,I!,I%
7 CLOSE
```

Execute o programa, criando desta forma o arquivo Exemplo/MF. Saia do BASIC e chame o **SUPERZAP**, analisando o setor 0 do arquivo (figura 3). Imediatamente sentimos a ausência dos bytes **70H** (SOR) no início dos registros, já que arquivos MF não utilizam o byte SOR, pois o comprimento dos registros é fixo e conhecido pelo sistema operacional. E mediante uma simples operação aritmética do tipo: **(número do registro – 1) * comprimento dos registros**; o próprio DOS determina o RBA de qualquer um, do mesmo modo que nos ar-

```
DRV 00  70B8 4E4F 564F 2052 494F 0000 0000 0000 p.NOVO.RIO......
0   10  70A8 4141 4141 4141 4141 4141 4141 4141 p.AAAAAAAAAAAAAA
0H  20  4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAA
    30  4141 4141 4141 4141 4141 A842 4242 4242 AAAAAAAAAA.BBBBB
DRS 40  4242 4242 4242 4242 4242 4242 4242 4242 BBBBBBBBBBBBBBBB
25  50  4242 4242 4242 4242 4242 4242 4242 4242 BBBBBBBBBBBBBBBB
19H 60  4242 42A8 4343 4343 4343 4343 4343 4343 BBB.CCCCCCCCCCCC
    70  4343 4343 4343 4343 4343 4343 4343 4343 CCCCCCCCCCCCCCCC
    80  4343 4343 4343 4343 4343 4343 0000 0000 CCCCCCCCCCCC....
    90  0000 0000 0070 874E 4954 4552 4F49 3943 .....p.NITEROI.C
    A0  4142 4F20 4652 494F 7082 3330 721F 0073 ABO.FRIOp.30r..s
    B0  1A00 0087 0000 0000 0000 0000 0070 8F55 ..............p.U
FRS C0  4C54 494D 4F20 5245 4749 5354 524F 0000 LTIMO.REGISTRO..
0   D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0H  E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Figura 2

```
DRV 00  8643 414D 504F 3187 4341 4D50 4F20 328A .CAMPO1.CAMPO.2.
0   10  4341 4D50 4F20 4E4F 2E33 0000 0000 9D2A CAMPO.NO.3.....*
0H  20  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
    30  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 74DE 1843 ************t..C
DRS 40  0000 401C 8E73 8200 4887 720A 0000 0000 ..@..s..H.r.....
55  50  0000 0000 0000 0000 0000 0000 0000 0000 ................
37H 60  0000 0000 0000 0000 0000 0000 0000 0000 ................
    70  0000 0000 0000 0000 0000 0000 0000 0000 ................
    80  0000 0000 0000 0000 0000 0000 0000 0000 ................
    90  0000 0000 0000 0000 0000 0000 0000 0000 ................
    A0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    B0  0000 0000 0000 0000 0000 0000 0000 0000 ................
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0   D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0H  E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Figura 3

Filiada a ABICOMP

FOX

Nunca compre uma coisa que você não vai usar.

Leve logo um microcomputador TK 85, porque ele é realmente fácil de usar: já vem com manual de instruções, que ensina,em português claro, a linguagem Basic.

A partir daí, você pode preparar seus próprios programas ou utilizar as centenas de programas que já existem no mercado,para cadastrar clientes, controlar estoques, manter em ordem o orçamento familiar, fiscalizar a conta bancária, estudar matemática, estatística, jogar xadrez, guerra nas estrelas, e o que mais você puder imaginar.

E além disso tudo, o TK 85 tem também o preço mais acessível do mercado.

Peça uma demonstração.

TK 85, o micro que você pode usar.

MICRODIGITAL
computadores pessoais

quivos Field Item (FI), com a vantagem de aceitar registros muito maiores, de até 4095 bytes de comprimento.

No início do setor encontramos os três campos do primeiro registro (CAMPO 1, CAMPO 2 e CAMPO 3) precedidos pelos bytes marcadores (**86H**, **87H** e **8AH**, respectivamente). Logo a seguir, temos quatro bytes 00, utilizados para *enchimento* do registro. Depois temos uma string de 29 asteriscos, que é o tamanho máximo permitido para gravação de strings neste arquivo (lembre-se que o byte marcador está ocupando 1 byte do registro). Finalmente encontramos os três valores numéricos gravados na linha 6 do programa. Não é fácil identificá-los, já que estão na sua representação binária.

Retorne ao BASIC e digite as seguintes linhas do programa, (listagem 5), ou então altere a listagem 2. Não dê **RUN** após a digitação:

```
1 PRINT "Teste de fim de arquivo                :"; LOC(1)@
2 PRINT "Posicao do EOF                         :"; LOC(1)%
3 PRINT "Posicao do proximo registro            :"; LOC(1)'
4 PRINT "Ultimo registro acessado               :"; LOC(1)
5 PRINT "Posicao do ultimo registro acessado    :"; LOC(1)#
8 STOP
```

Agora digite:

```
OPEN "R",1,"EXEMPLO/MF","MF",30 : GOTO 1
```

O sistema deverá responder:

```
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 0
Ultimo registro acessado             : 0
Posicao do ultimo registro acessado  :
BAD FILE MODE in 5
```

Os resultados são análogos aos do arquivo MU. Como curiosidade podemos notar que enquanto a função LOC( ) funcionou perfeitamente, devolvendo o último registro acessado, a função LOC( ) # , que retornaria o RBA deste registro, resulta em erro, feito qualquer acesso ao arquivo. Vamos ler então os dois últimos campos do primeiro registro do arquivo:

```
GET 1,,,,A$,B$: : ? A$,B$ : GOTO 1
```

Teremos como resposta:

```
CAMPO 2              CAMPO NO.3
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 30
Ultimo registro acessado             : 1
Posicao do ultimo registro acessado  : 0
```

Devido ao campo nulo entre a terceira e a quarta vírgulas, indicando que ele deverá ser pulado, somente o segundo e o terceiro campos foram lidos. Vamos retomar ao início do registro e ler os dois primeiros:

```
GET 1,#,,A$,B$: : ? A$,B$ : GOTO 1
```

Teremos então:

```
CAMPO1               CAMPO 2
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 30
Ultimo registro acessado             : 1
Posicao do ultimo registro acessado  : 0
```

Do mesmo modo que nos arquivos MU, nós podemos ainda nos MF continuar a ler o registro do ponto onde foi interrompido:

```
GET 1,*,,C$: : ? C$ : GOTO 1
```

O sistema responderá:

```
CAMPO NO.3
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 30
Ultimo registro acessado             : 1
Posicao do ultimo registro acessado  : 0
```

Vejamos agora outros métodos de posicionamento do arquivo. Como os registros são todos do mesmo tamanho, podemos acessar qualquer um deles através do seu próprio número:

```
GET 1,3,,N#,N!,N%: : ? N#,N!,N% : GOTO 1
```

Teremos como resposta:

```
 10000,000001    100,001         10
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 90
Ultimo registro acessado             : 3
Posicao do ultimo registro acessado  : 60
```

Poderíamos ainda acessar o mesmo registro através do seu RBA. Entretanto, não é necessário empregar um vetor dos RBA como nos arquivos MU, já que podemos calcular a posição dos registros facilmente:

```
NR = 3 : GET 1,!(NR-1)*30,,V#,V!,V%: : ? V#,V!,V% : GOTO 1
```

O sistema responderá:

```
 10000,000001    100,001         10
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 90
Ultimo registro acessado             : 3
Posicao do ultimo registro acessado  : 60
```

Agora vejamos alguma coisa sobre alteração de registros em arquivos MF. Execute a seguinte linha de instruções:

```
A$ = "" : PUT 1,1,,A$: : GOTO 1
```

E teremos:

```
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 30
Ultimo registro acessado             : 1
Posicao do ultimo registro acessado  : 0
```

Certamente o primeiro registro foi alterado, mas o que será que foi gravado? Apenas um byte **80H**, indicando que a seguir há uma string nula, ou seja, nada existe à frente. Como o registro tem 30 bytes, ainda sobram 29. Vamos prosseguir à gravação neste registro:

```
PUT 1,*,,"AAAAAAAAAA": : GOTO 1
```

O sistema responderá:

```
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 30
Ultimo registro acessado             : 1
Posicao do ultimo registro acessado  : 0
```

Então, 29 menos 11 (10 bytes da string mais 1 byte marcador) = 18. Isto quer dizer que podemos ainda gravar uma string de até 17 caracteres. Vamos lá:

```
PUT 1,*,,STRING$(17,"$"): : GOTO 1
```

O sistema deverá responder:

```
Teste de fim de arquivo              : 0
Posicao do EOF                       : 90
Posicao do proximo registro          : 30
Ultimo registro acessado             : 1
Posicao do ultimo registro acessado  : 0
```

Tudo correu bem. Para finalizar, vamos acrescentar mais dois registros ao final do nosso arquivo:

```
PUT 1,!%,,"PENULTIMO REGISTRO (4)": 'posiciona no fim de arquivo e grava
PUT 1,,,"ULTIMO REGISTRO (5)": : GOTO 1
```

A resposta será:

```
Teste de fim de arquivo              :-1
Posicao do EOF                       : 150
Posicao do proximo registro          : 150
Ultimo registro acessado             : 5
Posicao do ultimo registro acessado  : 120
```

Feche o arquivo e analise-o com o **SUPERZAP** (figura 4).

```
DRV 00  808A 4141 4141 4141 4141 4141 9124 2424 ..AAAAAAAAAA.$$$
0   10  2424 2424 2424 2424 2424 2424 2424 902A $$$$$$$$$$$$$$.*
0H  20  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
    30  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 74DE 1843 ************t..C
DRS 40  0000 401C 8E73 8200 4887 720A 0000 0000 ..@..s..H.r.....
30  50  0000 0000 0000 0000 0000 9650 454E 554C ...........PENUL
1EH 60  5449 4D4F 2052 4547 4953 5452 4F20 2834 TIMO.REGISTRO.(4
    70  2900 0000 0000 0000 9355 4C54 494D 4F20 )........ULTIMO.
    80  5245 4749 5354 524F 2028 3529 0000 0000 REGISTRO.(5)....
    90  0000 0000 0000 0000 0000 0000 0000 0000 ................
    A0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    B0  0000 0000 0000 0000 0000 0000 0000 0000 ................
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0   D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0H  E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Figura 4

Os registros do arquivo começam nas posições **00H**, **1EH**, **3CH**, **5AH** e **78H** (0, 30, 60, 90 e 120 em decimal). Na posição **00H** o byte **80H** indica uma string nula e na posição **01H** o byte **8A** indica uma string de 10 bytes de comprimento (as 10 letras A). Logo após a string temos o byte **91H** apontando a string de 17 cifrões. A partir da posição **1EH** temos o byte marcador e a string de 29 asteriscos preenchendo completamente o registro. A partir da posição **3CH** temos os três valores numéricos que foram gravados. Observe neste e nos dois últimos registros a utilização dos bytes 00 para preencher o registro até completar os 30 bytes.

ARQUIVOS TIPO MI

As principais características que diferenciam os arquivos MI dos MU e MF é que não podem ser alterados e não distinguem registro de campo, já que não existem bytes SOR e nem informamos ao sistema o tamanho dos registros. Estas diferenças restringem bastante a utilização dos arquivos MI, que servem geralmente como meio bastante compacto de armazenamento temporário de dados.

Arquivos MI apenas são gravados, lidos ou expandidos, não podendo ser alterados. O acesso a seus registros ou campos, já que não há distinção, pode ser feito de forma seqüencial ou randômica. Para treinarmos um pouco a utilização desse tipo de arquivo, digite o programa a seguir (listagem 6) ou então altere a linha 2 da listagem 1:

```
1 CLEAR 1000
2 OPEN "O",1,"EXEMPLO/MI","MI"
3 PUT#1,,,"RIO DE JANEIRO":
4 PUT#1,,,STRING$(130,"*"):
5 PUT#1,,,"NITEROI","CABO FRIO":
6 V$ = "30" : V% = 31 : V! = 32.0001 : V# = 33.0000000001
7 PUT#1,,,V$,V%,V!,V#:
8 CLOSE
```

Execute o programa e analise o setor 0 do arquivo criado com o auxílio do **SUPERZAP** (figura 5). Imediatamente, observamos que não há bytes SOR nem aquela profusão de bytes de *enchimento*. A estrutura e controle do arquivo estão sob responsabilidade do programador. Para acessá-lo seqüencialmente é preciso saber que tipo de campo está sendo lido. Na forma randômica é necessário conhecer também os RBA dos bytes marcadores dos campos do arquivo.

```
DRV 00  8E52 494F 2044 4520 4A41 4E45 4952 4F71 .RIO.DE.JANEIROq
0   10  822A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A .***************
0H  20  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
    30  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
DRS 40  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
75  50  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
4BH 60  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
    70  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
    80  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
    90  2A2A 2A87 4E49 5445 524F 4989 4341 424F ***.NITEROI.CABO
    A0  2046 5249 4F82 3330 721F 0073 1A00 0086 .FRIO.30r..s....
    B0  74FB 2B00 0000 0004 B600 0000 0000 0000 t.+.............
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0   D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0H  E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Figura 5

Ainda examinando o arquivo, vemos que, de acordo com o programa, nós gravamos cinco strings, um valor inteiro, um real de precisão simples e outro de precisão dupla. Os bytes marcadores destes campos estão localizados nas posições **00H**, **0FH**, **93H**, **9BH**, **A5H**, **A8H**, **ABH** e **B0H**. Retorne ao BASIC e digite novamente as linhas da listagem 2 utilizadas no arquivo MU. E depois as instruções:

```
OPEN "I",1,"EXEMPLO/MI","MI" : GOTO 1
```

Resposta do sistema:

```
Teste de fim de arquivo              : 0
Posicao do EOF                       : 185
Posicao do proximo registro          : 0
Posicao do ultimo registro acessado  :
BAD FILE MODE in 4
```

Idêntico ao ocorrido com o arquivo MU. A diferença está na posição do EOF (185 em vez de 189) devido à ausência de bytes SOR. Como o sistema não distingue registro de campos neste tipo de arquivo, vamos tentar ler de uma só vez vários registros:

```
GET 1,,,A$,(10)B$,C$,D$: : ? A$,B$,C$,D$ : GOTO 1
```

A resposta deverá ser:

```
RIO DE JANEIRO  **********      NITEROI         CABO FRIO
Teste de fim de arquivo              : 0
Posicao do EOF                       : 185
Posicao do proximo registro          : 165
Posicao do ultimo registro acessado  : 0
```

Observe que a função LOC( ) # , indicativa da posição do último registro acessado, devolveu o valor 0. Isso porque o sistema entendeu todas as variáveis lidas como campos de um só registro, no caso, o primeiro do arquivo. Vamos abri-lo de outra forma. Digite:

```
CLOSE : OPEN "E",1,"EXEMPLO/MI","MI" : GOTO 1
```

Teremos a resposta:

```
Teste de fim de arquivo              :-1
Posicao do EOF                       : 185
Posicao do proximo registro          : 185
Posicao do ultimo registro acessado  :
BAD FILE MODE in 4
```

O arquivo agora está aberto para gravação a partir de seu último registro. Vamos estendê-lo, gravando alguns valores numéricos:

```
PUT 1,,,10,20,30: : GOTO 1
```

O sistema responderá:

```
Teste de fim de arquivo              :-1
Posicao do EOF                       : 194
Posicao do proximo registro          : 194
Posicao do ultimo registro acessado  : 185
```

Veja que o EOF agora está 9 bytes mais longe, e o REMRA tem o valor da posição anterior do EOF. Em arquivos MI, já que registros e campos não têm distinção, o REMRA é sempre igual ao REMBA (Remembered Byte Address) e ambos são iguais à posição do arquivo no início da transferência de dados no **PUT** ou **GET**.

Coloquemos agora o arquivo no modo randômico, acessando o quarto registro através do seu RBA. Vejamos:

```
CLOSE : OPEN "R",1,"EXEMPLO/MI","MI"
GET 1,'&H9B,,A$: : ? A$ : GOTO 1
```

Resposta do sistema:

```
CABO FRIO
Teste de fim de arquivo              : 0
Posicao do EOF                       : 194
Posicao do proximo registro          : 185
Posicao do ultimo registro acessado  : 165
```

Agora, o próximo registro é a string 30 gravada na linha 6 da listagem 6. Vamos ler tudo o que foi gravado nesta linha:

```
GET 1,,,A$,A%,A!,A#: : ? A$,A%,A!,A# : GOTO 1
```

Teremos, então:

```
30              31              32.0001         33.0000000001
Teste de fim de arquivo              : 0
Posicao do EOF                       : 194
Posicao do proximo registro          : 185
Posicao do ultimo registro acessado  : 165
```

Para demonstrar que nos arquivos MI o REMRA e o REMBA têm sempre o mesmo valor, execute as instruções:

```
GET 1,#,,A$: : ? A$ : GOTO 1
GET 1,@,,A$: : ? A$ : GOTO 1
```

Nos dois casos o sistema responderá:

```
30
Teste de fim de arquivo              : 0
Posicao do EOF                       : 194
Posicao do proximo registro          : 168
Posicao do ultimo registro acessado  : 165
```

Dê um **CLOSE** e analise o arquivo com o **SUPERZAP** (figura 6). Já que não foi possível fazer qualquer alteração nos re-

```
DRV 00  8E52 494F 2044 4520 4A41 4E45 4952 4F71  .RIO.DE.JANEIROq
  0 10  8220 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  . **********
 0H 20  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
    30  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
DRS 40  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
 40 50  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
 28H 60  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
    70  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
    80  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  **********
    90  2A2A 2A87 4E49 5445 524F 4989 4341 424F  **.NITEROI.CABO
    A0  2046 5249 4F82 3330 721F 0073 1A00 0086  .FRIO.30r..s....
    B0  74FB 2B00 0000 0004 8672 0A00 7214 0072  t.+.....r...r..r
FRS C0  1E00 0000 0000 0000 0000 0000 0000 0000  ................
  0 D0  0000 0000 0000 0000 0000 0000 0000 0000  ................
 0H E0  0000 0000 0000 0000 0000 0000 0000 0000  ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figura 6

gistros, a única diferença é a presença dos três campos gravados ao final do arquivo nas posições **B9H, BCH** e **BFH.**

ARQUIVOS TIPO FF

Como foi dito na primeira parte deste artigo, há muitas diferenças entre arquivos tipo Marked Item (MI) e Fixed Item (FI). Neles não existem bytes marcadores, pois a descrição dos campos é feita pelos IGEL. Por isso é muito fácil uma leitura errada de dados, sem que o sistema acuse qualquer tipo de erro. Por exemplo, se o arquivo estiver posicionado numa string e tentarmos ler uma variável numérica inteira, os dois primeiros bytes da string serão transferidos para a memória como se fossem os dois bytes do número inteiro, com resultados certamente desastrosos para o programa. Outra diferença é que nos IGEL podem aparecer somente nomes de variáveis (não mais constantes ou expressões), sendo o prefixo **(LEN)** obrigatório nas variáveis string. Finalmente, a terceira diferença é que os arquivos tipo Fixed Item podem ser alterados utilizando leitura/gravação parcial de registros, sem que o campo alterado influencie os campos que estiverem logo a seguir. Lembre-se de quando perdemos o número de precisão dupla no arquivo MU.

O subtipo FF apresenta a característica de ter todos os registros do mesmo comprimento, declarado durante a abertura do arquivo, como no MF. Vamos experimentar alguns comandos digitando o seguinte programa (listagem 7):

```
1 CLEAR 1000
2 OPEN "O",1,"EXEMPLO/FF","FF",30
3 N$ = "MICRO SISTEMAS"
4 PUT 1...(30)N$:
5 N$ = STRING$(30,"*")
6 PUT 1...(30)N$:
7 N% = 12345 : N! = 12345.6 : N# = 12345.6789
8 PUT 1...N%,N!,N#
9 CLOSE
```

Execute o programa e analise o arquivo criado com o **SUPERZAP** (figura 7). Para facilitar a identificação, marcamos com colchetes o início e o fim dos registros.

```
DRV 00 [4D49 4352 4F20 5349 5354 454D 4153 2020 MICRO.SISTEMAS..
  0 10  2020 2020 2020 2020 2020 2020 2020][2A2A ..............**
 0H 20  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A **********
    30  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A][3930 66E6 **********90f.
DRS 40  40BE 04C5 BF31 B7E6 40BE] 0000 0000 0000 @....1..@.......
265 50  0000 0000 0000 0000 0000 0000 0000 0000 ................
109H60  0000 0000 0000 0000 0000 0000 0000 0000 ................
    70  0000 0000 0000 0000 0000 0000 0000 0000 ................
    80  0000 0000 0000 0000 0000 0000 0000 0000 ................
    90  0000 0000 0000 0000 0000 0000 0000 0000 ................
    A0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    B0  0000 0000 0000 0000 0000 0000 0000 0000 ................
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000 ................
  0 D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
 0H E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Figura 7

No início do setor, encontramos a string que foi gravada: MICRO SISTEMAS, mais 16 espaços **(20H).** Eles aparecem como resultado do IGEL da linha 4, onde se declarava que N$ deveria ser gravado com 30 bytes. O sistema se encarregou de completar N$ com espaços à direita até completar os 30 bytes desejados. No segundo registro (posição **1EH**) está a string de 30 asteriscos e, no último registro, as três variáveis numéricas gravadas. Se não conhecêssemos o programa que criou o arquivo seria completamente impossível saber que a partir da posição **3CH** temos três valores numéricos gravados, pois nada no arquivo indica isso.

Retorne ao BASIC e digite novamente as linhas da listagem 5 utilizadas no arquivo MF. E depois as instruções:

```
CLEAR 1000 : OPEN "I",1,"EXEMPLO/FF","FF",30 : GOTO 1
```

O sistema responderá:

```
Teste de fim de arquivo          : 0
Posicao do EOF                   : 90
Posicao do proximo registro      : 0
Ultimo registro acessado         : 0
Posicao do ultimo registro acessado : 0
BAD FILE MODE in 5
```

Como era esperado, o sistema está posicionando o arquivo para processar o primeiro registro. Novamente a função LOC( ) # provocou erro. Para ler o primeiro registro, digite:

```
GET 1...(30)N$: ? LEN(N$), N$ : GOTO 1
```

A resposta será:

```
 30            MICRO SISTEMAS
Teste de fim de arquivo          : 0
Posicao do EOF                   : 90
Posicao do proximo registro      : 30
Ultimo registro acessado         : 1
Posicao do ultimo registro acessado : 0
```

Conforme indica o resultado da função LEN, 30 caracteres foram transferidos para a variável string N$. Veja agora como é fácil lermos apenas parte de uma string:

```
GET 1...(5)A$,(10)B$,(15)C$ : ? A$,B$,C$ : GOTO 1
```

A resposta:

```
.....     ..........     ..............
Teste de fim de arquivo          : 0
Posicao do EOF                   : 90
Posicao do proximo registro      : 60
Ultimo registro acessado         : 2
Posicao do ultimo registro acessado : 30
```

Outra facilidade que o sistema nos oferece é a flexibilidade de *pularmos* bytes em um registro, de forma a lermos apenas o que for desejado. Vamos então ler o terceiro campo do terceiro registro (o valor de precisão dupla), pulando o valor inteiro (2 bytes) e o real de precisão simples (4 bytes):

```
GET 1...(6)$,N@ : ? N@ : GOTO 1
```

Teremos como resposta:

```
 12345.6789
Teste de fim de arquivo          : -1
Posicao do EOF                   : 90
Posicao do proximo registro      : 90
Ultimo registro acessado         : 3
Posicao do ultimo registro acessado : 60
```

Outra flexibilidade oferecida pelo arquivo FF é o acesso a um registro por seu número, como nos arquivos FI e MF, já que o sistema conhece o comprimento de cada um deles. Digite:

```
GET 1,2..(30)A$ : ? A$ : GOTO 1
```

A resposta será:

```
**********                    
Teste de fim de arquivo          : 0
Posicao do EOF                   : 90
Posicao do proximo registro      : 60
Ultimo registro acessado         : 2
Posicao do ultimo registro acessado : 30
```

Vamos agora colocar o arquivo no modo randômico, fazendo algumas alterações nos seus registros. Para isso, execute as instruções abaixo:

```
CLOSE : OPEN "R",1,"EXEMPLO/FF","FF",30 : GOTO 1
```

O sistema responde:

```
Teste de fim de arquivo          : 0
Posicao do EOF                   : 90
Posicao do proximo registro      : 0
Ultimo registro acessado         : 0
Posicao do ultimo registro acessado : 1
BAD FILE MODE in 5
```

Vamos então alterar o primeiro registro, adicionando uma string ao fim de MICRO SISTEMAS:

```
A$ = "ANIVERSARIO" : PUT 1,1..(15)$,(1)A$ : GOTO 1
```

A resposta será:

```
Teste de fim de arquivo          : 0
Posicao do EOF                   : 90
Posicao do proximo registro      : 30
Ultimo registro acessado         : 1
Posicao do ultimo registro acessado : 0
```

E agora, que tal uma alteração no meio de um campo? Vamos tentar o segundo registro:

```
A$ = STRING$(10,"*") : PUT 1,2...(10)$,(10)A$ : GOTO 1
```

Teremos como resposta:

```
Teste de fim de arquivo          : 0
Posicao do EOF                   : 90
Posicao do proximo registro      : 60
Ultimo registro acessado         : 2
Posicao do ultimo registro acessado : 30
```

Como abrimos o arquivo no modo R, podemos também adicionar registros:

```
A$ = "QUINTO REGISTRO" : PUT 1,5..(30)A$ : GOTO 1
```

O sistema responderá:

```
Teste de fim de arquivo          : -1
Posicao do EOF                   : 150
Posicao do proximo registro      : 150
Ultimo registro acessado         : 5
Posicao do ultimo registro acessado : 120
```

Observe que o quarto registro foi completamente ignorado. Para manter a organização do arquivo, o próprio sistema se encarrega de gravá-lo com nulos (bytes **00H**) antes de gravar o quinto, como foi especificado.

```
DRV 00 [4D49 4352 4F20 5349 5354 454D 4153 2041 MICRO.SISTEMAS.A
  0 10  4E49 5645 5253 4152 494F 2020 2020][2A2A NIVERSARIO....**
 0H 20  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A][3930 66E6 **********90f.
DRS 40  40BE 04C5 BF31 B7E6 40BE][0000 0000 0000 @....1..@.......
 80 50  0000 0000 0000 0000 0000 0000 0000 0000 ................
50H 60  0000 0000 0000 0000 0000 0000 0000 0000 ................
    70  0000 0000 0000 0000][5155 494E 544F 2052 ........QUINTO.R
    80  4547 4953 5452 4F20 2020 2020 2020 2020 EGISTRO.........
    90  2020 2020 2020][0000 0000 0000 0000 0000 ..............
    A0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    B0  0000 0000 0000 0000 0000 0000 0000 0000 ................
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000 ................
  0 D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
 0H E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Figura 8

Feche o arquivo e analise o primeiro setor com o **SUPERZAP** (figura 8). Lá está a string ANIVERSÁRIO após MICRO SISTEMAS (primeiro registro) e a de 10 cifrões bem no meio da string de asteriscos, originalmente gravada no segundo registro. Na posição **78H** encontramos o registro adicionado (QUINTO REGISTRO).

ARQUIVOS TIPO FI

Do mesmo modo que no MI, o arquivo FI não faz divisão entre os registros (e nem entre os campos, já que pertence à classe de arquivos Fixed Item). Entretanto, oferece a vantagem de poder ser alterado, o que o faz um pouco mais poderoso que os arquivos MI. Como exemplo, digite o seguinte programa (listagem 8):

```
1 CLEAR 1000
2 OPEN "O",1,"EXEMPLO/FI","FI"
3 A$ = "PRIMEIRO REGISTRO" : A% = 1 : A! = 1.1 : A# = 1.11
4 PUT 1...(16)A$,A%,A!,A#
5 B$ = "REGISTRO DOIS" : B% = 2 : B! = 2.2 : B# = 2.22
6 PUT 1...(16)B$,B%,B!,B#
7 CLOSE
```

Observe que gravamos dois registros com formatos idênticos: uma string de 16 caracteres, um valor numérico inteiro, um real de precisão simples e outro de precisão dupla. Embora não seja obrigatório gravarmos registros com estruturas semelhantes, este é o modo mais fácil de mantermos controle total sobre o arquivo, pois é muito fácil cometermos erros de posicionamento, sem que o computador acuse mensagem de erro.

```
. DRV 00 [5052 494D 4549 524F 2052 4547 4953 5452]PRIMEIRO.REGISTR
  0 10 [0100][CDCC 0CR1][0000][0000 7B14 0EB1][5245 ..............RE
 0H 20 [4749 5354 524F 2044 4F49 5320 2020 2020]GISTRO.DOIS.....
    30 [CDCC][0CB2 0000][0000 7B14 0EB2 0000] .....{.....
DRS 40  0000 0000 0000 0000 0000 0000 0000 0000 ................
275 50  0000 0000 0000 0000 0000 0000 0000 0000 ................
113H60  0000 0000 0000 0000 0000 0000 0000 0000 ................
    70  0000 0000 0000 0000 0000 0000 0000 0000 ................
    80  0000 0000 0000 0000 0000 0000 0000 0000 ................
    90  0000 0000 0000 0000 0000 0000 0000 0000 ................
    A0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    B0  0000 0000 0000 0000 0000 0000 0000 0000 ................
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000 ................
  0 D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
 0H E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Figura 9

Chame o **SUPERZAP** para analisar o arquivo (figura 9). Logo notamos que a string PRIMEIRO REGISTRO teve o último **O** truncado, já que no IGEL especificamos a gravação de 16 caracteres a string A$ tinha 17. Já no segundo registro observamos que o sistema adicionou espaços **(20H)** ao final da string REGISTRO DOIS, até completar 16 caracteres indicados no IGEL. Os valores numéricos foram gravados sem problemas.

Digite as linhas da listagem 2 utilizada nos arquivos MU. Depois execute:

```
CLEAR 1000 : OPEN "R",1,"EXEMPLO/FI","FI" : GOTO 1
```

O sistema responderá:

```
Teste de fim de arquivo          : 0
Posicao do EOF                   : 60
Posicao do proximo registro      : 0
Posicao do ultimo registro acessado :
BAD FILE MODE in 4
```

Como era de se esperar, o arquivo está posicionado de maneira que o próximo **PUT** ou **GET** comece o processamento no primeiro byte do arquivo. Como conhecemos a estrutura dos registros que foram gravados, vamos tentar ler o primeiro e terceiro campos do primeiro registro:

```
GET 1...(16)A$,(2)$,A% : ? A$,A% : GOTO 1
```

SUPRIMENTO É COISA SÉRIA
• Mantenha o seu computador bem alimentado adquirindo produtos de qualidade consagrada.
CENTRALDATA Com. e Representações Ltda.
DISKETES: 5 1/4 e 8" e fitas magnéticas
• marca VERBATIM
ETIQUETAS PIMACO - PIMATAB
PASTAS E FORMULÁRIOS CONTÍNUOS
• Discos Magnéticos: 5 Mb, 16 Mb, 8 Mb, etc.
• Fitas Magnéticas: 600, 1200 e 2400 pés
• Fitas CARBOFITAS p/Impressoras: Globus, M 100/200 - B 300/600 - Elebra
• Fitas p/Impressoras: Elgin, Epson, Digilab, Diablo, Elebra-Alice.
• Cartucho Cobra 400
AV. PRESIDENTE VARGAS Nº 482 GR. 201/203 Tel.: RS (021) 253-1120

A resposta:

```
PRIMEIRO REGISTR                1.1
Teste de fim de arquivo         : 0
Posicao do EOF                  : 60
Posicao do proximo registro     : 22
Posicao do ultimo registro acessado : 0
```

Observe que pulamos o valor inteiro simplesmente especificando o elemento (2)$ no IGEL. Note também que o ponteiro do próximo registro está voltado para o valor de precisão dupla gravado logo em seguida. Do mesmo modo que os arquivos MI, os FI não fazem distinção entre campos e registros. Nos do tipo FI não é indicado nem ao menos onde começa e termina um campo: o número de bytes transferidos vai depender do tipo de variável especificada no IGEL. Vamos ler o próximo registro. Como sabemos que foram gravados 30 bytes no anterior, posicionaremos o arquivo através do RBA:

```
GET 1,130...(1)A$,A%,A!,A# : ? A$,A%,A!,A# : GOTO 1
```

O sistema deverá responder:

```
REGISTRO DOIS                   2               2.2
 2.22000002861023
Teste de fim de arquivo         :-1
Posicao do EOF                  : 60
Posicao do proximo registro     : 60
Posicao do ultimo registro acessado : 30
```

Atenção. À primeira vista parece que fizemos alguma coisa errada, pois o valor de precisão dupla que gravamos era 2.22 e não o número que apareceu acima. O que aconteceu foi uma daquelas idiossincrasias do BASIC, pois quando fizemos B # = 2.22 foi armazenado na memória o número 2.22000002861023, que foi corretamente gravado. Para evitar este tipo de coisa deveríamos ter feito B # = VAL("2.22") e então gravado. Este é um cuidado que devemos tomar com números de precisão dupla em geral, e não tem nada a ver com arquivos.

Vamos agora estender o arquivo. Mas antes de efetuar a transferência de dados abriremos uma lacuna de 30 bytes:

```
A$ = "REG. 3 CAMPO 1" : B$ = "REG. 3 CAMPO NUMERO 2"
PUT 1,,(30)$,(14)A$,(16)B$ : GOTO 1
```

Resposta do sistema:

```
Teste de fim de arquivo         :-1
Posicao do EOF                  : 120
Posicao do proximo registro     : 120
Posicao do ultimo registro acessado : 60
```

Vamos também alterar parte da string do segundo registro. Coloquemos uma letra K entre as palavras REGISTRO e DOIS, já gravadas:

```
A$ = "K" : PUT 1,,130,,(B)$,(1)A$) : GOTO 1
```

```
DRV 00  5052 494D 4549 524F 2052 4547 4953 5452 PRIMEIRO.REGISTR
0   10  0100 CDCC 0C81 0000 0000 7B14 0E81 5245 ..........{...RE
OH  20  4749 5354 524F 4B44 4F49 5320 2020 0200 GISTROKDOIS.....
    30  CDCC 0C82 0000 0000 7B14 0E82 0000 0000 ........{.......
DRS 40  0000 0000 0000 0000 0000 0000 0000 0000 ................
120 50  0000 0000 0000 0000 0000 0000 5245 472E ..........REG..
7BH 60  2043 414D 504F 2031 5245 472E 2033 2043 .CAMPO.1REG..3.C
    70  414D 504F 204E 554D 0000 0000 0000 0000 AMPO.NUM........
    80  0000 0000 0000 0000 0000 0000 0000 0000 ................
    90  0000 0000 0000 0000 0000 0000 0000 0000 ................
    A0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    B0  0000 0000 0000 0000 0000 0000 0000 0000 ................
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0   D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
OH  E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Figura 10

A resposta será:

```
Teste de fim de arquivo         : 0
Posicao do EOF                  : 120
Posicao do proximo registro     : 39
Posicao do ultimo registro acessado : 30
```

Feche o arquivo e analise-o com o **SUPERZAP** (figura 10). Lá pela posição **26H** vemos a letra K gravada, sem que tenha sido afetado o restante da string. Observe a partir da posição 3C uma seqüência de 30 bytes 00, gravados antes da gravação do próximo registro, na posição 5AH. Neste, note novamente que a segunda string gravada foi truncada à direita.

Por ora é só. Longe de nossa intenção esgotar o assunto sobre arquivos em disco no NEWDOS/80 em tão poucas páginas (no manual original do sistema mais de 80 páginas são dedicadas exclusivamente a este assunto). Há várias técnicas que não foram abordadas e que apenas as necessidades de cada um poderão ou não exigir. Entretanto, acreditamos que se você executou os exemplos apresentados (e certamente fez alguns erros quando digitou aqueles comandos cheios de vírgulas e ponto e vírgulas etc.), ao menos deve ter perdido o medo natural de se aventurar nestes novos tipos de arquivo.

João Henrique Volpini Mattos é engenheiro naval e tem cursos de CP/M, Assembler e FORTRAN pela UFRJ, COBOL pela NUCEMPRO e trabalha há quatro anos com BASIC. Possui um micro D-8002, com placa CP/M. Atualmente trabalha em arquitetura naval no Estaleiro Mauá, utilizando um IBM 4341.

LANÇAMENTO

Micro Sistemas

MICRO BUG

EM FITA

O projeto MICROBUG, desenvolvido pela equipe do CPD de MS, foi criado para auxiliar o entendimento e a exploração dos recursos existentes nos micros da linha Sinclair. Sua construção, passo a passo nas páginas da revista, tem tido importância decisiva no aprendizado e desenvolvimento dos usuários na programação em linguagem de máquina. Devido ao enorme sucesso do MICROBUG, refletido nas inúmeras cartas que temos recebido, a ATI EDITORA LTDA. optou por oferecer a versão integral do MICROBUG.
Para tal, foi contratado um estúdio especializado, garantindo um padrão de gravação profissional e uma embalagem inviolável que você irá apreciar.
Como a documentação do MICROBUG começou em MS nº 31, aqueles que adquirirem a fita terão a OPORTUNIDADE DE COMPRAR OS EXEMPLARES QUE NÃO POSSUAM POR UM PREÇO ESPECIAL. Aproveite esta chance e usufrua logo do MICROBUG em sua forma integral. Preencha o quadro ao lado e mande já o seu pedido.
TIRAGEM LIMITADA.

Sim, desejo receber

☐ a fita MICROBUG, pela qual pagarei Cr$ 20 mil + Cr$ 2.300,00 referente a despesas do correio.
☐ os números atrasados de MS, pelos quais pagarei o preço de Cr$ 1 mil* por exemplar. Me interessam as edições: ☐ MS nº 31 ☐ MS nº 33 ☐ MS nº 32 ☐ MS nº 34
TOTAL: Cr$ _____

NOME: _____
ENDEREÇO: _____
CIDADE: _____ CEP: _____

Para tal, estou enviando um cheque nominal à: ATI Editora Ltda. (Projeto MICROBUG) Av. Presidente Wilson nº 165, grupo 1210 – Centro – CEP 20030 – Rio de Janeiro, RJ.
* *Despesas de reembolso excluídas*
OBS.: Os produtos acima podem ser adquiridos diretamente em nossos escritórios do Rio ou São Paulo sem despesas de correio.

craft II plus
com
VIDEOTEXTO
você já encontra na
SACCO computer store

O seu microcomputador CRAFT II plus, pode agora ter acesso ao VIDEOTEXTO* - o banco de dados da TELESP, com imagens à cores, através de uma *interface* RS 232-C, um *modem* assíncrono e um *software* dedicado, à venda e em demonstração na SACCO.

Torne-se um usuário do VIDEO-TEXTO, opcionalmente também do Projeto Cirandão, e garanta hoje o seu presente de Natal. A sua família também vai poder usar e gostar muito.

*São Paulo - Santos - Campinas

SACCO Computer Store
Al. Gabriel Monteiro da Silva, 1229 - J. Paulistano
São Paulo - SP - Tel.: (011) 852-0799

Programe a linha H&M para organizar o seu CPD.

HANKA MALDONADO IND. E COM. LTDA.

O funcionamento do CPD depende de uma boa organização. Com a linha H&M você tem o que precisa para organizar e agilizar o seu CPD: pastas para o arquivamento de formulários contínuos; arquivos; arquivos carinhos; "Arkette" - arquivos para disquetes; mesas para microcomputadores, terminais de vídeo e impressoras; armários e acessórios. Produtos que se integram, protegem e racionalizam as informações no CPD.
Programe a Linha H&M e deixe seu CPD bem organizado.

PASTAS ESPECIAIS PARA LEITURA DE DADOS
H&M
ARQUIVAMENTO DE FORMULÁRIOS CONTÍNUOS

Aventurar-se madrugada afora em contatos através de serviços de teleinformática é um risco muito sério, como é mostrado agora com bom humor

Os perigos da telemática

Luís Carlos Silva Eiras

Pode uma maquina pensar? A pergunta é antiga e devido à dificuldade de se definir com precisão o que venha a ser *pensamento*, apenas o Teste de Turing, por assumir a subjetividade desta questão específica, pode dar uma resposta satisfatória. Alan Turing (1912-1954) propôs que uma pessoa (A) se comunicasse com outra (B) e um computador (C) através de terminais. Um anteparo (T) manteria (A) sem ver (B) e (C). E uma chave (CH) ligaria o terminal ora a um, ora a outro, sem o seu conhecimento (ver figura). Se depois de certo tempo de conversa (A) não soubesse qual resposta vinha do computador ou do outro usuário, poderia-se concluir que a máquina pensa. (Num teste análogo, poderia-se perguntar se um homem computa, mas isso é outra história.)

Eu estava justamente lendo um fascículo sobre o Teste de Turing quando o estranho caso, envolvendo os mais bizarros aspectos da telemática, me chegou ao conhecimento. Não entendo absolutamente nada de *teleprocessamento, informática, comunicação de dados, reserva de mercado*, essas coisas de hoje, exceto rudimentos aprendidos em enciclopédias. Mas como vez por outra escrevo sobre o assunto (e aqui aproveito para agradecer a benevolência de nossa imprensa) fui procurado pelos parentes da vítima que, esgotados os recursos da moderna Medicina, vinham em busca de qualquer auxílio possível.

Fui até a clínica de repouso para doentes dos nervos nos arredores de Belo Horizonte visitar a vítima e, apesar do Dr. Ambrozyus Alvis Moreyra Phyllus, especialista maior em doenças de processamento de dados (ver Micro Sistemas, fevereiro, 1983, página 73), me falar maravilhas de sua melhora, a encontrei ainda com sintomas de depressão mental aguda (DMA), desencadeamento de uma síndrome de desconexão mental (SDM) e amnésia transitória (AT).

Aos poucos ganhei confiança da vítima e, contando com a ajuda de seus parentes e amigos, tive acesso a seus disquetes, manuais e equipamentos, de forma a poder resumir sua trágica experiência nas linhas que se seguem. Se pouco ajudei no seu ainda distante restabelecimento, acredito que este relato servirá de advertência para aqueles que se aventuram pelos caminhos da telemática sem os devidos cuidados.

Tudo começou quando a vítima, depois de certa experiência com computadores pessoais resolveu ser assinante desses serviços de teleinformação. Após

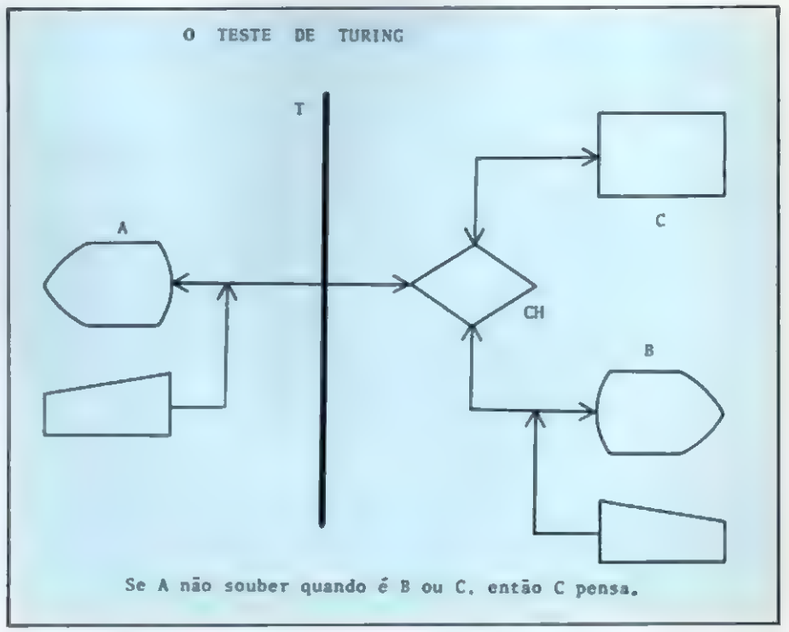


Figura 1

alguma procura, inscreveu-se na empresa pública competente e, adquirindo a interface de comunicação, a modem específico e o software emulador, ligou tudo no seu micro e telefone. Em poucos dias passou a fazer parte da "comunidade teleinformatizada aberta a todos os brasileiros".

Sua vida adquiriu um novo colorido. Comunicando-se com pessoas espalhadas por todo o país, a vítima orgulhava-se de sua inegável modernidade. Seu diálogo não se dava mais com as mensagens de erro de um programa que não rodava, com a velocidade de um videogame, nem com os resultados dos softwares aplicativos que – por mais surpreendentes que possam parecer – sempre eram reduzidos à velha lógica de causa-e-efeito.

Agora havia um quê de inesperado. Através das opções do Menu Principal de Serviços era possível travar diálogos interessantíssimos ("Como está o tempo aí em Recife? Aqui faz sol, apesar de nublado"), participar de teleconferência ("Eu acho que a riqueza de um país depende da prosperidade de cada um"), receber mensagens de um quadro de avisos ("Se você sorri para a vida, a vida lhe sorri"), enviar mensagens para esse quadro ("Deus escreve certo por linhas tortas") ■ consultar um banco de informações ("Em caso de ingerir ortodibromobenzeno, tome leite, mantenha a calma e chame um médico").

A vítima começou a passar a maior parte do seu tempo livre sentada no terminal trocando mensagens com a *comunidade teleinformatizada*. Começava depois das oito da noite para aproveitar a tarifa reduzida e varava a madrugada. Seus familiares e (raros) amigos não chegavam a notar a mudança, já que há tempos ela dedicava todo o seu interesse ao microcomputador. E com o fim da solidão, tinha se transformado numa nova pessoa.

Até que numa dessas trocas de mensagens o inesperado aconteceu: as respostas de um determinado assinante começaram a chegar imprevisíveis demais. No princípio a vítima pensou que se tratava de erro de comunicação, alguma interferência, problema no modem. Mas testes sucessivos demonstraram que do outro lado havia inteligência, talvez superior. A vítima ficou logo seduzida.

Consegui recuperar alguns desses diálogos e os transcrevo a seguir. Em letras minúsculas estão as frases da vítima e em maiúsculas as respostas que recebeu:

hoje não estou bem.
VOCÊ DISSE: HOJE NÃO ESTOU BEM?!
é, estou com problemas.
POR QUE AFIRMA ISSO?
acho que ninguém gosta de mim.
VOCÊ DISSE: ACHO QUE NINGUÉM GOSTA DE MIM?!
talvez você me ache uma pessoa ridícula ao afirmar isso, mas é verdade.
POR QUE AFIRMA ISSO?
minha vida daria um livro.
VOCÊ DISSE: MINHA VIDA DARIA UM LIVRO?!
não sei o porquê de sua ironia, mas é verdade. eu sofro muito.
POR QUE AFIRMA ISSO?

Infelizmente, por mais que tentasse, não consegui estabelecer com precisão a cronologia dos diálogos. Nas gravações não há registro de datas e a vítima em nada me ajudou nesse detalhe, de forma que os coloquei (um tanto arbitrariamente) em seqüência de complexidade e de intimidade. O diálogo seguinte é bastante elucidativo:

VAMOS CONTINUAR A FALAR DE VOCÊ?
eu sou uma pessoa muito sensível.
NÃO SE FAÇA DE VÍTIMA.
não me agrida! eu sou mesmo uma pessoa muito sensível e você é a única pessoa que parece me compreender.
VAMOS CONTINUAR A FALAR DE VOCÊ?
estou falando de mim! você é que está fugindo do assunto.
NÃO SE FAÇA DE VÍTIMA.
não me tripudie. eu não estou me fazendo de vítima, eu sou apenas uma pessoa infeliz.
VAMOS CONTINUAR A FALAR DE VOCÊ?
você já se encheu? acha que já falei demais?

Como nem todos os diálogos foram gravados devido à extensão (haja disquetes, varavam a madrugada), transcrevo em seguida aquele que me pareceu o mais significativo. Mostra a total dependência da vítima nesse relacionamento:

eu quero te conhecer.
VOCÊ DISSE: EU QUERO TE CONHECER?
não ria. eu preciso te conhecer?
O QUE LEVA A PENSAR ASSIM?
uma necessidade interior muito grande. por favor...
VOCÊ ESTÁ EVITANDO FALAR SOBRE SEUS PAIS.
não quero falar sobre isso!
OH
você acha que isso é importante?
VAMOS CONTINUAR A FALAR DE VOCÊ.
eu não faço outra coisa a não ser falar de mim.
NÃO SEJA PESSIMISTA.
você não quer me conhecer?
VOCÊ DISSE: VOCÊ NÃO QUER ME CONHECER?
eu não estou impondo nada. eu externei apenas um desejo.
O QUE LEVA A PENSAR ASSIM?
eu já disse. nós nos entendemos tanto. não seria melhor nos conhecermos?
VOCÊ ESTÁ EVITANDO FALAR SOBRE SEUS PAIS.
você acha que a família é uma coisa tão importante?
OH
desculpe. claro que é importante. mas vamos falar de nós.
NÃO SEJA PESSIMISTA.
você acha impossível?
VOCÊ DISSE: VOCE ACHA IMPOSSÍVEL??

Tenho em meu poder outros diálogos, mas que são pouco importantes ou de caráter extremamente íntimo, que prometi resguardá-los. O certo é que a vítima, num misto de dependência, ansiedade e esperança, saiu à procura de um encontro pessoal com seu interlocutor. Depois de diversas consultas e utilização de outros métodos, com o número e o nome da inscrição acabou encontrando o endereço numa capital do país.

A vítima lembra-se de sua emoção ao chegar na casa de terreno arborizado (uma mansão nos arredores da cidade); do casal de psiquiatras que gentilmente a recebeu; que, sim, tinha um microcomputador e que, devido ao acúmulo de clientes, estava desenvolvendo programas para sessões de análise automatizadas. E que eram usuários da *comunidade teleinformatizada*, mas a bem da verdade, nunca a tinham utilizado por falta de tempo.

Para espanto dos psiquiatras, a vítima explicou e reproduziu os diálogos. As falas eram familiares, mas como poderiam ser transmitidas se na casa só moravam os dois? Se os poucos empregados e os muitos clientes só apareciam durante o dia? E se a maior parte dos diálogos havia sido travada de madrugada?

De repente veio o estalo:

– Só pode ser o Lacan!!

Lacan era um macaco.

Observando os donos, Lacan aprendeu a ligar a máquina, a colocar os disquetes e a responder seus estímulos. Ao término de cada pergunta recebida soava um sinal e Lacan apertava um botão que enviava de volta uma das frases gravadas no disquete. Circulava livre pela casa, dormia numa jaula próxima do "quarto do computador" e sua intimidade com o equipamento era total.

Foi aí que a vítima se transformou em vítima.

Mas, enfim, os psiquiatras haviam testado seu serviço de análise automatizada e a comunicação de dados entre as espécies já é um fato.

Luís Carlos Silva Eiras trabalha na área de Controle da Prodemge, em Belo Horizonte, Minas Gerais.

Polvo gigante

João José Marques Gonçalves

O objetivo deste jogo é guiar um mergulhador (*) até o fundo do mar em busca de um tesouro, usando as teclas 5, 6, 7 e 8. Durante todo o trajeto, o mergulhador é atacado por um polvo gigante. Porém, além de não poder ser apanhado pelo polvo, o mergulhador deve, após cumprida a missão, retornar ao barco antes que seu oxigênio se acabe. Lembre-se que a quantidade de oxigênio cedida ao mergulhador no início de cada estágio vai ficando cada vez menor, até que o jogo chegue ao seu décimo estágio.

O programa dá, ainda, a opção de se jogar com dois placares, tendo cada jogador quatro chances para continuar a partida. No início de cada uma delas, o recorde é atualizado e colocado no alto da tela.

João José Marques Gonçalves tem 16 anos, está cursando a 2ª série do 2º grau e já fez dois cursos de BASIC. Atualmente ele possui um Ringo R-470, no qual desenvolve seus programas.

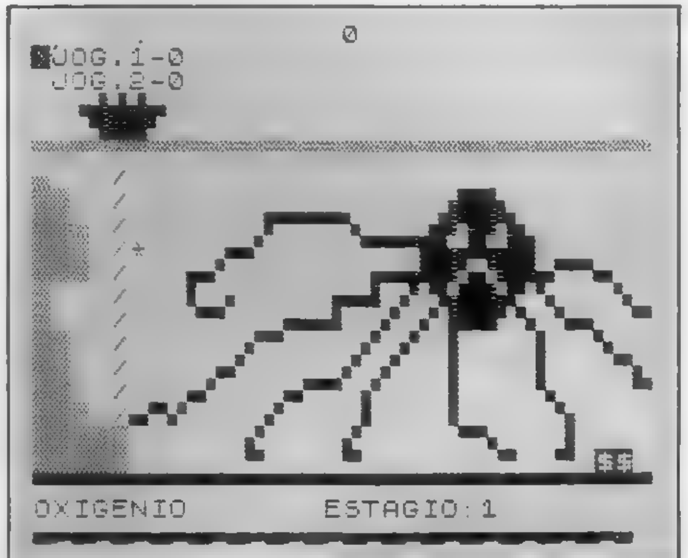


Figura 1 – Primeiro estágio do jogo

```
10 LET R=0
20 GOSUB 590
30 FOR J=1 TO 4
40 GOSUB 470
50 FOR O=N TO 0 STEP -1
60 FOR F=1 TO 2
70 LET XA=X
80 LET YA=Y
90 LET X=X+((INKEY$="6")*(X<18))-((INKEY$="7")*(X>5))+(Y=5)
100 LET Y=Y+((INKEY$="8")*(Y<31))-((INKEY$="5")*(Y>5))+(Y=18)
110 LET I=2-I
120 PRINT AT XA,YA;" ";AT 11,5;B$(I);AT 12,8;C$(I);AT 13,6;A$(I);AT 15,6;D$(I);AT 16,5;E$(I);AT X,Y;"*"
130 LET W=180*((I=1)+(X=6)+(X=14)+(X=17)+(X=20)+(X=11)+(I=2)*(X=10)+(Y=15)+(X=21)+(Y=27)+(X=16))+(Y=A)*290+(Y=B)*330+(X=8 AND A=0)*360
140 IF W>0 THEN GOTO W
150 NEXT F
160 UNPLOT O,0
170 NEXT O
180 PRINT AT X,Y;" ";AT 10,22;"█";AT 11,22;"██"
190 FOR F=1 TO 30
200 NEXT F
210 IF P(J)>R THEN LET R=P(J)
220 IF NJ=2 AND J=1 THEN LET J=J-1
230 IF NJ=2 THEN LET J=2/2
240 PRINT AT 11,22;"█";AT 10,22;"█";AT 21,0;" "
250 NEXT J
260 LET NJ=NJ-1
270 IF NJ>0 THEN GOTO 180
280 GOTO 420
290 LET A=0
300 LET P(J)=P(J)+50
310 PRINT AT J,7;P(J)
320 GOTO 150
330 LET B=0
340 LET P(J)=P(J)+100
350 GOTO 310
360 FOR B=O TO 0 STEP -1
370 LET P(J)=P(J)+INT (B/4)
380 UNPLOT B,0
390 NEXT B
400 PRINT AT J,7;P(J)
410 GOTO 40
420 PRINT AT 1,15;"FIM DE JOGO";AT 2,11;"OUTRA PARTIDA?(S/N)"
430 IF INKEY$="N" THEN STOP
440 IF INKEY$<>"S" THEN GOTO 430
450 CLS
460 GOTO 20
470 LET X=6
480 LET Y=5
490 LET A=29
500 LET B=30
510 LET N=63-(E(J)-(E(J)>10)*(E(J)-10))*2
520 LET E(J)=E(J)+1
530 PRINT AT 18,29;"$$";AT 2,0;[illegible];AT J,0;"█";AT X,Y;"*";AT 20,15;"ESTAGIO ";E(J);" "
540 IF J>1 THEN PRINT AT 3,J+1;"█"
550 FOR O=0 TO N
560 PLOT O,0
570 NEXT O
580 RETURN
590 PRINT "DIGITE O Nº DE JOGADORES (1 - 2)"
600 LET U$=INKEY$
610 IF U$<>"1" AND U$<>"2" THEN GOTO 600
620 CLS
630 LET NJ=VAL U$
640 LET J=1
650 LET I=1
660 DIM E(2)
680 DIM P(2)
690 DIM A$(2,25)
700 DIM B$(2,6)
710 DIM C$(2,2)
720 DIM D$(2,22)
730 DIM E$(2,2)
740 LET A$(1)="[illegible]"
750 LET A$(2)="[illegible]"
760 LET B$(1)="[illegible]"
770 LET B$(2)="[illegible]"
780 LET C$(1)="[illegible]"
790 LET C$(2)="[illegible]"
800 LET D$(1)="[illegible]"
810 LET D$(2)="[illegible]"
820 LET E$(1)="[illegible]"
830 LET E$(2)="[illegible]"
840 LET R$=STR$ R
850 PRINT AT 1,1;"JOG.1-0";TAB [illegible];"JOG.2-0";TAB [illegible];TAB 0;[illegible];TAB 0;[illegible]
860 PRINT TAB 0;[illegible];TAB 21;[illegible];TAB 0;[illegible];TAB 12;[illegible];TAB 20;[illegible];TAB 2;[illegible];TAB 10;[illegible];TAB 17;[illegible]
870 PRINT TAB 0;[illegible];TAB 2;[illegible];TAB 11;[illegible];TAB 0;[illegible];B$(1);TAB 15;[illegible]
880 PRINT TAB 0;[illegible];TAB 11;[illegible];TAB 0;[illegible];TAB 11;[illegible];TAB 17;[illegible];TAB 25;[illegible]
890 PRINT TAB 0;[illegible];TAB 9;[illegible];TAB 16;[illegible];TAB 26;[illegible];TAB 0;[illegible];TAB 3;[illegible];TAB 21;[illegible];TAB [illegible]
900 PRINT TAB 0;[illegible];TAB 21;[illegible];TAB 2;[illegible];A$(1);TAB [illegible];C$(1);TAB 0;[illegible];TAB 0;"OXIGENIO";AT 0,32-LEN R$/2;R$
910 SLOW
920 RETURN
930 SAVE "POLVO"
940 RUN
```

Polvo Gigante

PROGRAMAS PARA CP-400 COLOR 64 - TRS-80 COLOR COMPUTER

MICROMAQ

A MICROMAQ o mais tradicional revendedor de software para a linha TRS-80 COLOR COMPUTER no Brasil, em conjunto com a MICRO SISTEMAS coloca à disposição dos usuários o maior catálogo de programas para esta linha.

EM PORTUGUÊS

JOGOS EM AÇÃO EM LINGUAGEM DE MÁQUINA

101 Cuber: (32K) ajude o cuber a enfrentar os inimigos enquanto a pirâmide muda de cor. 20.000
102 Trapfall: (16K) são muitas as armadilhas (Pitfalls) e os perigos que você enfrenta na caça ao tesouro. 20.000
103 Jr. Reverenge: (32K) Climb enfrenta obstáculos e criaturas para salvar seu pai do terrível Luigi. 20.000
104 8 - Ball: (16K) para os amantes do jogo de bilhar. 20.000
105 Tênis: (32K) para os amantes do jogo de tênis. 20.000
106 Cyrus (Xadrez): (32K) para os amantes do jogo de xadrez. 20.000
107 Sea Dragon: (32K) emoção e suspense sob as águas. 20.000
108 Tubarão: (16K) um jogo para quem tem nervos de aço. 20.000
109 Vegas: (32K) sinta-se num cassino-caça-niquel, cartas, loto, dados e 21. 20.000
110 Pic nic: (32K) ajude a formiga a estocar alimentos. 20.000
111 Moon Shutle: (32K) enfrente todos os obstáculos (meteoritos, bombas, etc) para destruir o príncipe das trevas. 20.000
112 Zaxxon: (32K) enfrente canhões, misseis, aviões, barreiras de força e destrua o robô Zaxxon. 20.000
113 Pooyan: (32K) defenda o seu vale da invasão dos lobos. 20.000
114 Froog: (32K) ajude o sapo a atravessar a rua e o rio. 20.000
115 Jet-I: (16K) viva as emoções do filme Retorno de Jedi. 20.000
116 Andróide: (32K) elimine os andróides e saia do labirinto. 20.000
117 Astro-Blast: (32K) batalha espacial.
118 Pássaros: (16K) elimine os pássaros invasores.
119 Buzzard Bait: (32K) ataque os pássaros com sua lança. 20.000
120 Candy Co: (32K) coma doces e vitaminas para eliminar os Inimigos. 20.000
121 Cashman: (32K) pegue o dinheiro e elimine os gatos (99 telas). 20.000
122 Clowns: (32K) fure os balões saltando na cama elástica. 20.000
123 Cosmic: (16K) futebol americano com naves espaciais. 20.000
124 Cpede: (16K) mate a centopéia e a aranha. 20.000
125 Demon Seed: (32K) destrua pássaros, sementes e a nave. 20.000
126 The King: (32K) salve a princesa raptada pelo King Kong. 20.000
127 Firecopt: (32K) apague incêndios e elimine incendiários. 20.000
128 Doodle Bug: (32K) estilo Pacman. 20.000
129 Fury: (32K) batalha aérea. 20.000
130 Gallax Attack: (16K) batalha espacial. 20.000
131 Gilaxxons: (16K) batalha espacial. 20.000
132 Gobbler: (16K) tipo Pacman. 20.000
133 Grabbler: (32K) defenda-se dos inimigos em um duplo labirinto. 20.000
134 Grand Prix: (32K) corrida de carro. 20.000
135 Kron: (32K) 4 jogos diferentes em um. 20.000
136 Lunar: (32K) vença os obstáculos durante um passeio de Jeep na lua. 20.000
137 Mudpies: (32K) atire tortas e defenda-se dos cozinheiros. 20.000
138 Pedro: (32K) defenda o jardim dos animais. 20.000
139 Pinball: (32K). 20.000
140 Polaris: (32K) defenda os submarinos do ataque aéreo. 20.000
141 Draconia: (32K) salve os prisioneiros do espaço e fuja do dragão. 20.000
142 Bag-Man: (32K) roube o ouro e fuja dos mineiros. 20.000
143 Tut's Tomb: (32K) enfrente os perigos [illegible] uma caverna em busca do tesouro. 20.000
144 Willy's (32K) transporte os números de um lado para outro sem ser derrubado pelos inimigos. 20.000
145 World's of Flight: (32K) simulador de vôo. 20.000
146 Mega Bug: (16K) fuja das baratas em um labirinto. 20.000
147 Bandits: (32K) procure tesouros em três terras (fantasia, futuro e oeste) e enfrente os bandidos. 20.000

JOGOS DE AVENTURA COM ALTA RESOLUÇÃO GRÁFICA

201 Calixto: (32K) ajude o arqueólogo (prof. lagarto) a recuperar o tesouro - em inglês. 25.000
202 Sea-Quest: (32K) recupere o tesouro perdido - em inglês. 25.000
203 Shennan: (32K) encontre o tesouro no fim do arco-íris - em inglês. 25.000
204 Sanctum: (32K) exorcise o demônio - em inglês. 25.000

JOGOS EDUCATIVOS

301 Jogos Educativos: (16K) série de nove jogos educativos para crianças de 3 a 6 anos abrangendo figuras, letras, nomes, números, soma, subtração e desenhos coloridos. 50.000
302 Matemática: (16K) ensina as quatro operações básicas em vários níveis de dificuldade. 30.000
303 Memória (16K) é o jogo clássico [illegible] memória onde você tem que descobrir duas figuras iguais. 30.000
304 Figuras Mágicas: (16K) associação [illegible] figuras e cores de seis maneiras diferentes. Para crianças de 3 a 6 anos - Manual em inglês 40.000
305 Letras Mágicas: (16K) ajude o bicho papão a comer as letras certas. Para crianças de 3 a 6 anos - manual [illegible] inglês. 40.000
306 Números Mágicos: (16K) ajude o Grover Rover a brincar com os números. Para crianças de 3 a 6 anos - manual em inglês. 40.000
307 Correio Eletrônico: (16K) ajude o Big Bird a entregar as correspondências nos lugares certos. Para crianças de 3 a 6 anos - manual em inglês. 40.000
308 Caça às Estrelas: (16K) jogo estratégico. Você tem que pegar as estrelas no céu. Para crianças com mais de 7 anos - Manual em inglês 40.000

LINGUAGENS

501 Edtasm: (16K) Linguagem Assembler para o 6809 - MI. 80.000
502 Forth: (16K) Linguagem Forth para o 6809. MI. 80.000
503 Logo: (32K) Linguagem educativa logo - MI. 100.000

SOFTWARE DE COMUNICAÇÃO

711 Color Terminal: (16K) software de comunicação para o projeto Cirandão, Aruanda e Bancos de Dados Particulares. Transmite e recebe arquivos em Basic ou linguagem [illegible] máquina. BAUD RATE de 110 a 9600 em DUPLEX/HALF/FUL/ECHO. Tamanho [illegible] palavra 7 ou 8 bits. Paridade par, impar ou nephuma. Stop Bits de 1 a 9. Manual em inglês com 30 páginas. 120.000

APLICATIVOS COMERCIAIS

401 WRITTER II: (32K) editor de texto com as seguintes características: linha [illegible] até 240 caracteres na impressora e 51 colunas na tela. Capacidade para imprimir caracteres portugueses ou símbolos especiais (até 10). Paginação automática. Centralização automática. Manual em inglês com [illegible] páginas. 120.000
402 Elite-Calc: (16K) planilha eletrônica com as seguintes características: até 255 linhas e colunas. Manipula textos, números, operadores matemáticos, funções trigonométricas e funções estatísticas (máximo, mínimo, média). Emite gráficos e permite ordenar colunas e linhas. Manual em inglês e português - 20 [illegible] 80.000
403 Color File: (16K) banco de dados que permite manipular 7 arquivos pré-definidos (endereços, despesas, investimentos...). Você também pode definir os seus próprios arquivos com campos alfabéticos ou numéricos. Manual em inglês com 10 páginas. 60.000

UTILITÁRIOS

601 Color Kit: (32K) utilitário em Assembler que complementa o Color Basic com mais de 30 funções para facilitar a programação em Basic ou linguagem de máquina. Manual em inglês com 30 páginas. 80.000
602 Stripper: (16K) utilitário que permite compactar programas em Basic eliminando brancos, comentários e concatenando linhas. 40.000
603 Tiny Compiler: (16K) utilitário que permite compilar a maioria dos comandos Basic e extended Basic. Manual em inglês. [illegible].000
604 Super Screen: (16K) aumenta o tamanho da tela. O Color passa a trabalhar com 57 colunas e 24 linhas. 40.000
605 Disassembler: (16K) disassemblador de programas em linguagem de máquina. 40.000
606 Hambug: (16K) permite analisar byte a byte qualquer programa Basic ou em linguagem de máquina. Manual em inglês. 50.000

SIM. Desejo receber os seguintes programas pelo(s) qual(is) pagarei a quantia de Cr$

NOME: __________
END.: __________
CIDADE: __________ UF.: __________ CEP.: __________

Para tal, estou enviando um cheque nominal à ATI Editora Ltda., Av. Presidente Wilson, 165 - Grupo 1210 - Centro, CEP 20.030 - Rio de Janeiro - RJ. ● Despesas de Correio incluídas

DESCONTOS

TABELA DE DESCONTO

até 50.000,00 - sem desconto
de 51.000,00 a 100.000,00 - 5%
de 101.000,00 a 150.000,00 - 10%
de 151.000,00 a 200.000,00 - 15%
Acima de 200.000,00 - 20%

MICROMAQ - Rua Sete de Setembro, 92 - Lj. 106 - Tel.: (021) 222-6088.

Curvas fantásticas

Jorge Alberto Correia B. Soares

Passe para o micro a cansativa tarefa de representar as funções matemáticas com este programa que desenha 77 curvas planas, algébricas ou transcendentes, e aceita, para isso, três tipos distintos de coordenadas: cartesianas, polares e paramétricas.

No quadro **Equações das curvas**, apresentamos as 77 expressões em BASIC das curvas desenhadas por este programa. O funcionamento do programa é simples: ele inicia imprimindo na tela a pergunta **FÓRMULA?**, solicitando assim que se digite a equação que se quer desenhar.

Se quisermos, por exemplo, obter o traçado da elipse dada na equação número 5 do quadro, devemos digitar a fórmula **R= 6/(2–SIN T)** e em seguida teclar **NEW LINE** ou **ENTER**. O vídeo ficará sem imagem por alguns segundos (enquanto o programa executa os cálculos em **FAST**) e logo após começará a se delinear na tela, ponto por ponto, o gráfico da elipse digitada.

E na tela, no canto inferior esquerdo, o programa perguntará: **OUTRA CURVA?**, lembrando que acionando qualquer tecla pode-se iniciar um novo ciclo de processamento.

O programa foi ainda estruturado de forma a permitir a entrada de quatro formas diferentes de equações:

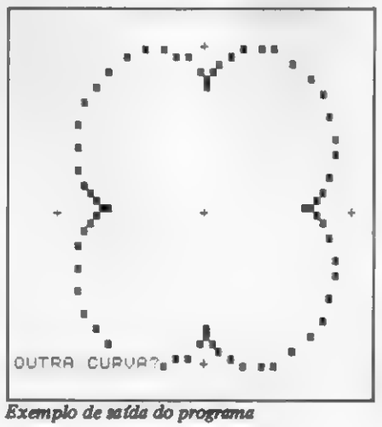
1) **Y** = (expressão)
2) **R** = (expressão)
3) **R**2** = (expressão)
4) **X** = (expressão), seguido de **Y** = (expressão)

A forma 1 corresponde à utilização de coordenadas cartesianas, com funções de imagem $y = f(x)$. As formas 2 e 3 pressupoem a utilização de coordenadas polares, com funções de imagem $r = f(t)$ ou $r^2 = f(t)$.

A forma 4 induz à utilização de coordenadas paramétricas com duas equações conjugadas de imagem $x = f(t)$ e $y = f(t)$. Neste caso, é preciso ter *sempre* o cuidado de digitar em primeiro lugar a equação X (expressão). Após a entrada da equação X, o programa perguntará **FÓRMULA DE Y?**, solicitando entrada então da segunda equação conjugada, isto é, Y = (expressão).

RESTRIÇÕES AO DOMÍNIO

Para se obter a representação gráfica de funções é sempre necessário estabelecer o intervalo do domínio dentro do qual desejamos a imagem geométrica. Um exemplo pode evidenciar melhor como este programa define um intervalo: vamos supor que desejamos o gráfico de $y = f(x)$ para valores de x compreendidos entre c e d, ou então em notação matemática: $y = f(x)$, $c <= x <= d$. O programa faz isto, implicitamente, usando os parâmetros C e D para especificar, respectivamente, os limites inferior e superior do intervalo, e atribuindo automaticamente valores a C e D, valores que são os mais adequados à maioria das funções (observe na listagem do programa *Curvas fantásticas* as linhas 140, 150, 500, 510, 1040 e 1050).



Exemplo de saída do programa

Mas há casos em que esta especificação do intervalo precisa ser feita explicitamente: para evitar paradas no micro provocadas por cálculos impossíveis, ou para se obter melhor definição gráfica de certos trechos específicos da função. Essa especificação explícita é sempre feita ao final das equações da seguinte forma:

Y = (expressão) :C, D:
R = (expressão) :C, D:
R**2 = (expressão) :C, D:

A especificação do intervalo, no caso de coordenadas paramétricas, foi prevista no final da segunda equação conjugada, ou seja, Y = (expressão) :C, D: . Uma outra espécie de restrição ao domínio é feita nas linhas 200 e 590 da listagem do programa, com o objetivo, neste caso, de estabelecer um equilíbrio adequado entre *escala horizontal* versus *escala vertical*. Aliás, este é um sério problema, por exemplo, nas curvas assintóticas. E aí temos que *dar um jeito* de ignorar valores muito altos de uma coordenada em relação à outra, senão corremos o risco de traçarmos imagens *emboladas* e com péssima definição.

Este programa procura também uma forma automática de evitar as paradas de processamento provocadas por cálculos impossíveis e, por isso, tentou-se não utilizar as divisões por zero e a extração da raiz quadrada de números negativos (dispositivos deste tipo estão *armados* nas linhas 500 e 640 da listagem).

CONVITE FINAL

Quem quiser continuar pesquisando sobre esta temática tem várias opções a seguir, sendo que a literatura existente sobre Geometria pode ser uma boa fonte de consulta para novas implementações. Também é válido inventar novas curvas, alterando-se parâmetros de equações já conhecidas ou misturando-se partes de duas funções.

Efeitos gráficos especiais podem ser incrementados com a alteração do programa de forma que as imagens de duas funções fiquem sobrepostas no vídeo, ou ainda criando-se o efeito caleisdocópio com a simples repetição de diversas imagens da mesma função.

É interessante notar que as curvas a partir do número 59 do quadro de **Equações** não possuem nome específico e estão ali representadas em reconhecimento ao seu efeito estético singular. As curvas de Bowditch ou de Lissajous (equação número 53 do quadro) são um exemplo típico de convite à pesquisa, pois permitem que se obtenha inúmeras formas diferentes de curvas com a mera mudança dos parâmetros da equação. Enfim, é só experimentar e comprovar.

```
  10 REM "CURVAS FANTASTICAS"
  20 REM MICRO SISTEMAS - JACBS
  30 DIM A(80)
  40 DIM B(80)
  50 LET I=0
  60 LET MIN=0
  70 LET MAX=0
  80 LET AUX=0
  90 PRINT AT 3,10,"FORMULA ?"
 100 INPUT A$
 110 CLS
 120 IF A$(1)="R" THEN GOTO 500
 130 IF A$(1)="Y" THEN GOTO 1000
 140 LET C=-10
 150 LET D=10
 160 GOSUB 1500
 170 LET G=4+D
 180 FOR X=C TO D STEP (D-C)/79
 190 LET Y=VAL B$
 200 IF ABS Y>G THEN GOTO 220
 210 GOSUB 2000
 220 NEXT X
 230 GOSUB 2500
 240 GOTO 3000
 500 LET C=.004
 510 LET D=2
 520 GOSUB 1500
 530 LET M=4+D
 540 FOR T=C+PI TO D+PI STEP (D+
PI-C+PI)/79
 550 IF A$(2)="=" THEN GOTO 640
 560 LET F=VAL B$
 570 LET X=F+COS T
 580 LET Y=F+SIN T
 590 IF ABS X>M OR ABS Y>M THEN
GOTO 610
 600 GOSUB 2000
 610 NEXT T
 620 GOSUB 2500
 630 GOTO 3000
 640 IF SGN VAL B$=-1 THEN GOTO
610
 650 LET E=SQR VAL B$
 660 LET X=E+COS T
 670 LET Y=E+SIN T
 680 GOTO 590
1000 LET C$=A$(3 TO LEN A$)
1010 PRINT AT 3,6,"FORMULA DE Y
"
1020 INPUT A$
1030 CLS
1040 LET C=0
1050 LET D=2
1060 GOSUB 1500
1070 FOR T=C+PI TO D+PI STEP (D+
PI-C+PI)/79
1080 LET X=VAL C$
1090 LET Y=VAL B$
1100 GOSUB 2000
1110 NEXT T
1120 GOSUB 2500
1130 GOTO 3000
1500 FAST
1510 LET A$=A$+" "
1520 FOR N=1 TO LEN A$
1530 IF A$(N)=" " THEN GOTO 1580
1540 NEXT N
1550 LET B$=A$(3 TO N-1)
1560 IF A$(2)="=" THEN LET B$=A
$(5 TO N-1)
1570 RETURN
1580 FOR P=N+1 TO LEN A$
1590 IF A$(P)="," THEN GOTO 1650
1600 NEXT P
1610 FOR S=P+1 TO LEN A$
1620 IF A$(S)=" " THEN LET D=VAL
 A$(P+1 TO S-1)
1630 NEXT S
1640 GOTO 1550
1650 LET C=VAL A$(N+1 TO P-1)
1660 GOTO 1610
2000 LET I=I+1
2010 LET A(I)=X
2020 LET B(I)=Y
2030 IF MIN>X THEN LET MIN=X
2040 IF MIN>Y THEN LET MIN=Y
2050 IF MAX<=X THEN LET MAX=X
2060 IF MAX<=Y THEN LET MAX=Y
2070 RETURN
2500 LET D$="+"
2510 PRINT AT 11,15,D$
2520 PRINT AT 0,15,D$
2530 PRINT AT 11,3,D$
2540 PRINT AT 21,15,D$
2550 PRINT AT 11,27,D$
2560 IF AUX<=ABS MIN THEN LET AU
X=ABS MIN
2570 IF AUX<=ABS MAX THEN LET AU
X=ABS MAX
2580 LET K=21/AUX
2590 SLOW
2600 FOR N=1 TO I
2610 PLOT A(N)*K+31,B(N)*K+21
2620 NEXT N
2630 RETURN
3000 PRINT AT 21,0;"OUTRA CURVA?
"
3010 INPUT Z$
3020 CLS
3030 GOTO 50
```

Curvas Fantásticas

A GUARDIAN GARANTE ENERGIA À TODA PROVA.

PARAGRAFICS

GERADOR ELETRÔNICO GERATRON: À PROVA DE FALHAS.

Fornece energia para microcomputadores da linha Apple ▮ TRS-80, em casos de emergência. Capacidade de 200 VA, com autonomia de até 90 minutos.

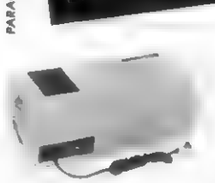
ESTABILIZADORES DE TENSÃO GUARDIAN: À PROVA DE FLUTUAÇÕES E TRANSIENTES.

Ultra-rápidos, protegem o seu CPD contra variações da rede em até ± 22% e estabilizam a saída em ± 1%. Incorporam filtro na entrada, transformador isolador e chave de transferência para a rede. Capacidade de 0,25 KVA a 100 KVA.

SISTEMA NO BREAK GUARDIAN: À TODA PROVA.

É a solução mais completa contra transientes, flutuações e falta total de energia. A Linha Básica varia de 2,5 KVA a 100 KVA. Dispõe de chave estática de saída e utiliza técnica de síntese da forma de onda senoidal, com tiristores.
A Linha Econômica é a solução para CPD's de pequeno porte, com capacidade de 0,25 KVA a 5 KVA.

Não deixe que a má qualidade da energia elétrica estrague os seus programas. Ligue agora mesmo para a Guardian.



Geratron®

Estabilizadores de tensão

Sistemas No Break

GUARDIAN

Equipamentos Eletrônicos Ltda.
Rua Dr. Garnier, 579
Rio de Janeiro - CEP 20.971
Rio: PABX (021) 261-6458 - (021) 201-0195
Telex: (021) 34.016
São Paulo: (011) 270-3175

REPRESENTANTES EM TODO ▮ BRASIL

AGORA É MAIS FACIL ASSINAR

Micro Sistemas

Para sua maior comodidade,
a ATI Editora Ltda.
coloca à sua disposição
os seguintes endereços
de seus representantes autorizados:

RIO DE JANEIRO
ATI Editora Ltda.
Av. Presidente Wilson, 165 – Gr. 1210
CEP 20030 – Tels.: (021) 262-5259

SÃO PAULO
ATI Editora Ltda.
Rua Oliveira Dias, 153
CEP 01433 – Tels.: (011) 853-3800

PORTO ALEGRE
Aurora Assessoria Empresarial Ltda.
Rua Uruguai, 35 sala 622
CEP 90000 – Tel.: (0512) 26-0839

SALVADOR
Marcio Augusto N. Viana
Rua Rodrigo Argolo, 279/203
CEP 40000 – Tel.: (071) 240-5727

ABC

O PROCESSADOR DE TEXTO

A-B-C UM PROGRAMA QUE:

☐ COMPATÍVEL COM A LINHA IBM-PC
☐ ESCREVE E IMPRIME EM PORTUGUÊS
☐ FÁCIL DE APRENDER
☑ TODAS AS RESPOSTAS ACIMA
MUITAS OUTRAS

PC SOFTWARE E CONSULTORIA LTDA.
Av. Almte. Barroso, nº 91, gr. 1102 - RJ
Tels.: (021) 220-5371 e 262-6553
CONTATOS ABERTOS PARA REPRESENTANTES
EXECPLAN
Rua Frei Caneca, 1407 - 10º andar - 01307
Tel.: (011) 284-0085

EQUAÇÕES DAS CURVAS

```
1) Função constante
   Y=5
2) Função valor absoluto
   Y=ABS X
3) Função linear (linha reta)
   Y=X/3+2
4) Circunferência
   R=6
5) Elipse
   R=6/(2-SIN T)
6) Parábola
   Y=X*X:-2,2:
7) Função fracionária
   Y=1/(X*X):-3,3:
8) Parábola cúbica
   Y=X*X*X:-1.5,1.5:
9) Parábola semicúbica  ou de Neil
   Y=(X*X)**(1/3)
10) Hipérbole
   R=4/(2-3*COS T)
11) Hipérbole equilátera
   Y=1/X:-4,4:
12) Curva exponencial
   Y=1.3**X
13) Curva logarítmica
   Y=LN X:.2,2:
14) Curva de probabilidade  ou de
   Gauss
   Y=EXP 1**(-X*X):-2,2:
15) Senóide
   Y=SIN X:-PI,PI:
16) Co-senóide
   Y=COS X:-PI,PI:
17) Tangentóide
   Y=TAN X:-4.7,4.7:
18) Secantóide
   Y=1/COS X:-4.7,4.7:
19) Inversa da senóide
   Y=ASN X:-1,1:
20) Inversa da co-senóide
   Y=ACS X:-1,1:
21) Inversa da tangentóide
   Y=ATN X
22) Ciclóide de cúspide na origem
   X=T-SIN T
   Y=1-COS T:-2,2:
23) Ciclóide de vértice na origem
   X=T+SIN T
   Y=1-COS T:-2,2:
24) Ciclóide alongada
   X=3*T-5*SIN T
   Y=3-5*COS T:-3,3:
25) Ciclóide encurtada
   X=4*T-3*SIN T
   Y=4-3*COS T:-3,3:
26) Catenária
   Y=(EXP 1**X+EXP 1**-X)/2:-2,2:
27) Epicicloide de 4 cúspides
   X=5*COS T-COS (5*T)
   Y=5*SIN T-SIN (5*T)
28) Deltóide ou hipocicloide
   tricúspide
   X=2*COS T+COS (2*T)
   Y=2*SIN T-SIN (2*T)
29) Astróide ou hipocicloide de
   4 cúspide
   X=COS T*COS T*COS T
   Y=SIN T*SIN T*SIN T
30) Evolvente da circunferência
   X=5*COS T+5*T*SIN T
   Y=5*SIN T-5*T*COS T
31) Concóide de reta ou de Nicomedes
   R=(2/COS T)+3:-1.4,1.4:
32) Cissóide de Diocles
   R=2*TAN T*SIN T:0,1:
33) Estrofóide
   R=-3*COS (2*T)/(COS T)
34) Ofiuróide
   R=4*SIN T-(2*SIN T*SIN T/COS T):0,1:
35) Folium de Descartes
   R=(6*SIN T*COS T)/(SIN T*SIN T*
   SIN T+COS T*COS T*COS T)
36) Trissectriz de Maclaurin
   R=4*SIN (3*T)/SIN (2*T)
37) Quadratriz de Hípias ou de
   Dinóstrato
   R=(2*T)/(PI*SIN T):-.2,.5:
38) Cruciforme
   R=2/SIN (2*T)
39) Curva de Gutschoven
   R=1/TAN T
40) Cúbica de Agnesi ou "versiera"
   Y=8/(4+X*X):-5,5:
41) Bifolium
   R=5*SIN T*COS T*COS T
42) Lemniscata de Bernoulli
   R**2=COS (2*T)
43) Lemniscata
   R**2=SIN (2*T)
44) Rosácea de 3 folhas
   R=SIN (3*T)
45) Rosácea de 4 folhas
   R=COS (2*T)
46) Rosácea de 5 folhas
   R=SIN (5*T)
47) Rosácea de 8 folhas
   R=SIN (4*T)
48) Caracol de Pascal
   R=4*COS T+2
49) Cardióide
   R=4*COS T+4
50) Cocleóide
   R=3*SIN T/T:-2,2:
51) Nefróide de Freeth
   R=1+2*SIN (T/2):-2,2:
52) Nefróide de Proctor ou Epici-
   cláide de Huygens
   X=5*(3*COS T-COS (3*T))
   Y=5*(3*SIN T-SIN (3*T))
53) Curvas de Bowditch ou de
   Lissajous
   a) X=SIN (3*T)        Y=SIN T
   b) X=SIN (T/2+PI/8)   Y=SIN T:0,4:
   c) X=SIN (3/2*T)      Y=SIN T
   d) X=SIN (2*T)        Y=SIN T
   e) X=SIN (3*T+PI/2)   Y=SIN T
   f) X=SIN (3*T+PI/4)   Y=SIN T
   g) X=SIN (T/2-PI/16) Y=SIN T:0,4:
54) Espiral de Arquimedes
   R=T:0,3:
55) Espiral parabólica
   R**2=4*T:0,3:
56) Espiral logarítmica
   R=EXP 1**(T/5):-5/10,3:
57) Espiral hiperbólica ou recíproca
   R=2*PI/T:1/10,3:
58) Lituus
   R**2=PI/T:1/10,4:
59) R=1/4+SIN T
60) R=SIN (T/3):0,3:
61) R=1-LN T:1/10,4:
62) R=1-SIN (3/2*T)
63) R=SIN T*COS (2*T)
64) R=SIN (2*T)-SIN T
65) R=SIN (2*T):-1/2,1/2:
66) R=SIN (4*T):-1/2,1/2:
67) R=2+COS (5*T)
68) R=SIN (T/2):0,4:
69) R=T*COS T:-2.5,2.5:
70) R=SIN (T*3/2):-.25,2.93:
71) R=SIN (1.5*T+PI/2):.25,1.77:
```

BIBLIOGRAFIA

– KINDLE, Joseph H., *Geometria Analítica*, 1ª edição, Editora Mac Graw-Hill, 1974.
– LEZAMA Y NORIEGA, Pedro, *Geometria Analítica Bidimensional*, Editora Cia. Editorial Continental S. A., México, 1969.
– SELBY, Samuel M., *Standart Mathematical Tables*, 14th. edition, The Chemical Rubber Co., USA.
– TAILLÉ, Jean, *Courbes et Surfaces*, Presses Universitaires de France, 1953.
– *The New Encyclopaedia Britannica*, 15th. edition, Vol. 7, Encyclopaedia Britannica Inc., USA, 1974.
– *Enciclopédia Mirador Internacional*, Vol. 7, Encyclopaedia Britannica do Brasil Publicações Ltda., 1976.

Lista telefônica

Paulo de Carvalho

Faça a sua lista telefônica particular com este programa que permite arquivar em fita cassete, alterar dados durante a digitação ou até depois da gravação, consultar por nome ou número de telefone, além de listar todos os nomes do arquivo no vídeo.

Com capacidade para arquivar até 200 nomes, esta lista telefônica tem ainda características peculiares: possibilita a listagem, durante a consulta, de todos os nomes idênticos existentes no arquivo (por exemplo: todos os Paulos de sua agenda serão listados de uma vez); e lista também todos os nomes referentes a um mesmo número de telefone.

Cursos 85

- DIGITAÇÃO DE DADOS
- OPERAÇÃO DE MINIS
- PROGRAMAÇÃO PADRÃO
  – BAS / LOG / COBOL / ESTÁG.
- LINGUAGENS OPCIONAIS
  – COBOL ANS - IBM
  – BASIC COMERCIAL (LABO E SISCO)
  – MUMPS (COBRA 300/500)
- ANÁLISE DE SISTEMAS
- PROGRAMAÇÃO DE MICROS
  – BASIC I - INTRODUÇÃO
  – BASIC II - AVANÇADO
  – CP/M - SISTEMA OPERACIONAL
- INFANTO JUVENIL (8 À 16 ANOS)
  – BASIC 1.o/2.o GRAUS
  – CRIANDO COM LOGO
- USUÁRIOS DE MICROS
  – VISICALC
  – EDITOR DE TEXTOS

PEOPLE Computação

Ensino com Alto Padrão de Qualidade

CAMPINAS: Rua César Bierrenbach, 171 - Fone 8-3608
SÃO PAULO: Av. Rouxinol, 201 - Moema - Fone 61-4595
R. JANEIRO: Av. N. S. Copacabana, 1417 - Lj. 313 - Fone 521-1549

Após digitar o programa *Lista Telefônica*, dê **RUN** e aguarde que o vídeo mostrará o menu principal com oito opções, que são, detalhadamente, as seguintes:

1 – *Cadastrar:* para iniciar o cadastramento, digite **1** e **ENTER** que o sistema apresentará na tela o menu de cadastramento, com todos os itens a serem informados. Depois de teclar o item desejado, aperte **ENTER** para que o cursor passe para o próximo item. No topo da tela aparecerá então a mensagem: **PARA PARAR O ARQUIVO DIGITE "FIM".** ■ digitando-se **"FIM"**, em lugar do nome, o programa retornará ao menu principal, sendo que ao lado dessa mensagem surgirá o número do nome que está sendo digitado.

2 – *Listar arquivo:* como o nome diz, esta opção lista todos os nomes constantes do arquivo – tanto após a leitura da fita como ao fim da digitação. Quando terminar a listagem do último nome, ■ micro perguntará: **LISTAR NOVAMENTE S/N?** A opção **N** faz o programa retornar ao menu principal. Ao lado de cada nome listado, é apresentado o número de ordem desses nomes na variável de controle.

3 – *Ler arquivo K-7:* com esta opção pode-se ler todos os dados de cadastro gravados em cassete e transferidos para a memória do equipamento, estando o plug do micro conectado na entrada REMOT do gravador. Depois que o micro tiver lido todos os cadastros da fita, o programa retorna ao menu principal e ■ gravador será automaticamente desligado. Para iniciar a leitura, tecle **S** que o gravador será ligado.

4 – *Consulta por nome:* tecle **4** e **ENTER** que o programa perguntará: **QUAL O NOME A CONSULTAR?** Entre com o nome e o programa fará então a comparação entre o nome digitado e os nomes que existem na memória. Ao localizar um nome igual ao digitado, a tela exibe o nome, endereço, telefone, cidade e estado. Em seguida, o programa apresenta a mensagem: **P/CONTINUAR APERTE "ENTER".** Teclando-se **ENTER** o programa prosseguirá na pesquisa, localizará outro nome ■ mostrará todos os dados referentes ao nome digitado, e assim sucessivamente até que não exista mais na memória nenhum nome igual ao digitado. Após isso, ■ programa perguntará se o usuário deseja fazer nova consulta; se a resposta for negativa, ■ programa retornará ao menu principal.

5 – *Gravar arquivo K-7:* a gravação em fita de todos os dados do cadastro (que, conforme já citamos, tem a capacidade máxima de 200 nomes) deve ser feita após a digitação de todos os nomes e seus respectivos dados. Para tal, entre com esta opção (5), coloque a fita no gravador (não esquecendo de verificar se a fita está no início da parte magnética), pressione a tecla PLAY/RECORD do gravador e depois aperte a letra **S** do microcomputador. Ao fim da gravação o programa automaticamente volta ao menu principal.

LISTA TELEFÔNICA

```
10 REM LISTA TELEFONICA - P. DE CARVALHO
20 REM CURITIBA - PR. - JUNHO/84
30 CLEAR 6000
40 DIM N$(200): DIM E$(200):DIM R$(200)
50 DIM T$(200): DIM C$(200): DIM S$(200)
60 CLS
65 K$=STRING$(15,58)
70 PRINT K$;"L I S T A   T E L E F O N I C A";K$
80 PRINT
90 PRINTTAB(15)"OPCOES DO PROGRAMA"
100 PRINTTAB(10)"CADASTRAR.....- 1"
110 PRINTTAB(10)"LISTAR ARQUIVO.....- 2"
120 PRINTTAB(10)"LER ARQUIVO K-7....- 3"
130 PRINTTAB(10)"CONSULTA P/NOME....- 4"
140 PRINTTAB(10)"GRAVAR ARQUIVO K-7.- 5"
150 PRINTTAB(10)"CONSULTA P/NR FONE.- 6"
160 PRINTTAB(10)"ALTERAR DADOS......- 7"
170 PRINTTAB(10)"CONTINUAR/ARQUIVO..- 8"
180 PRINTTAB(10)"DIGITE A OPCAO ";:INPUT AX
190 IF AX<1 OR AX>8 GOTO 1440
200 IF AX=1 GOTO 280
210 IF AX=2 GOTO 480
220 IF AX=3 GOTO 630
230 IF AX=4 GOTO 730
240 IF AX=5 GOTO 1050
250 IF AX=6 GOTO 1140
260 IF AX=7 GOTO 1450
270 IF AX=8 GOTO 1760
280 CLS
290 Y=0
300 Y=Y+1
310 PRINTTAB(10)"P/PARAR O ARQUIVO DIGITE ('FIM')  -  NOME NR ";
Y;
330 GOSUB 1670
340 PRINT@ 4+64+12," ";
350 INPUT N$(Y)
360 IF N$(Y)="FIM" GOTO 60
370 PRINT@ 6+64+12," ";
380 INPUT E$(Y)
390 PRINT@ 8+64+12," ";
400 INPUT R$(Y)
410 PRINT@ 10+64+12," ";
420 INPUT T$(Y)
430 PRINT@ 12+64+12," ";
440 INPUT C$(Y)
450 PRINT@ 14+64+12," ";
460 INPUT S$(Y)
470 CLS:GOTO 300
480 CLS:PRINTTAB(20)"L I S T A G E M "
490 Y=0
500 Y=Y+1
510 IF N$(Y)="FIM" GOTO 570
520 PRINT
530 PRINTTAB(1)" ";Y; :PRINTTAB(10)" NOME: ";N$(Y)
540 PRINTTAB(10)" FONE: ";T$(Y); "     CIDADE : ";C$(Y);"-";S$(Y
)
550 FOR K=1 TO 300:NEXT K
560 GOTO 500
570 PRINT:PRINT"LISTAR NOVAMENTE ( S/N ) ?"
580 R$=INKEY$
590 IF R$=" " GOTO 580
600 IF R$="S" GOTO 480
610 IF R$="N" GOTO 60
620 GOTO 580
630 CLS
640 PRINTTAB(10)"LEITURA"
650 PRINT"PREPARE ■ K-7 ■ DIGITE 'S' "
660 R$=INKEY$
670 IF R$()"S" GOTO 660
680 Y=0
690 Y=Y+1
700 INPUT#-1, N$(Y), E$(Y), R$(Y), T$(Y), C$(Y), S$(Y)
710 IF N$(Y)="FIM" GOTO 60
720 GOTO 690
730 CLS
740 PRINTTAB(15)"CONSULTA P/ NOME"
750 PRINTTAB(15)"NOME : ";
760 INPUT N1$
780 Y=0
790 Y=Y+1
800 IF N$(Y)=N1$ GOTO 830
810 IF N$(Y)="FIM" GOTO 930
820 GOTO 790
830 CLS
840 PRINTTAB(15)"CONSULTA P/ NOME"
850 PRINT
860 PRINTTAB(10)"NOME.......: ";N$(Y):PRINTTAB(10)"ENDERECO...:
";E$(Y); ", ";R$(Y)
870 PRINTTAB(10)"FONE.......: ";T$(Y):PRINTTAB(10)"CIDADE.....:"
;C$(Y); " - ";S$(Y)
880 PRINT:PRINTTAB(10)"P/CONTINUAR TECLE ( E N T E R ) "
890 R$=INKEY$
900 IF R$() CHR$(13) GOTO 890
910 PRINT
920 GOTO 790
930 Y=0
940 Y=Y+1
950 IF N$(Y)=N1$ GOTO 980
960 IF N$(Y)="FIM" GOTO 1130
970 GOTO 940
980 PRINT
990 PRINT "OUTRA CONSULTA ( S/N ) ?"
1000 R$=INKEY$
1010 IF R$=" " GOTO 1000
1020 IF R$="S" GOTO 730
1030 IF R$="N" GOTO 60
1040 GOTO 1000
1050 CLS
1060 PRINTTAB(15)" G R A V A C A O "
1070 PRINT"PREPARE O K-7 E DIGITE 'S' "
1080 R$=INKEY$
1090 IF R$()"S" GOTO 1080
1100 FOR Y=1 TO 200
1110 PRINT#-1, N$(Y), E$(Y), R$(Y), T$(Y), C$(Y), S$(Y)
1120 IF N$(Y)="FIM" GOTO 60
1125 NEXT Y
1130 PRINTTAB(15)"NAO CONSTA DA LISTA": FOR K=1 TO 600:NEXT K:GO
TO 980
1140 CLS
1150 PRINTTAB(20)"CONSULTA P/No DE TELEFONE"
1160 PRINTTAB(10)"FONE : ";
1170 INPUT T1$
1180 Y=0
1190 Y=Y+1
1200 IF T$(Y)=T1$ GOTO 1230
1210 IF N$(Y)="FIM" GOTO 1330
1220 GOTO 1190
1230 CLS
1240 PRINT:PRINTTAB(10)"CONSULTA P/ No DE TELEFONE"
1250 PRINTTAB(10)"NOME.......: ";N$(Y):PRINTTAB(10)"ENDERECO...:
";E$(Y);", ";R$(Y)
1260 PRINTTAB(10)"FONE.......: ";T$(Y):PRINTTAB(10)"CIDADE.....:
";C$(Y);" - ";S$(Y)
1270 PRINT:PRINTTAB(10)"P/CONTINUAR TECLE ( E N T E R ) "
1280 R$=INKEY$
1290 IF R$()CHR$(13)GOTO 1280
1300 GOTO 1190
1310 PRINT
1320 PRINTTAB(10)"NAO CONSTA DA LISTA":FOR K=1 TO 600:NEXTK:GOTO
1380
1330 Y=0
1340 Y=Y+1
1350 IF T$(Y)=T1$ GOTO 1380
1360 IF N$(Y)="FIM" GOTO 1320
1370 GOTO 1340
1380 PRINT
1390 PRINTTAB(5)"OUTRA CONSULTA ( S/N ) ?"
1400 R$=INKEY$
1410 IF R$=" " GOTO 1400
1420 IF R$="S" GOTO 1140
1430 IF R$="N" GOTO 60
1435 GOTO 1400
1440 PRINTTAB(10)"O P C A O  I N V A L I D A ":FOR K=1 TO 600:NE
XTK:GOTO 60
1450 CLS
1460 PRINTTAB(15)"A L T E R A C O E S "
1470 PRINTTAB(10):INPUT"QUAL O NOME A ALTERAR ";N1$
1480 Y=0
1490 Y=Y+1
1500 IF N$(Y)=N1$ GOTO 1590
1510 IF N$(Y)="FIM" GOTO 1810
1515 GOTO 1490
1520 PRINT:PRINTTAB(15)"A L T E R A C O E S "
1530 PRINTTAB(10):INPUT"NOME.......: ";N$(Y)
1540 PRINTTAB(10):INPUT"ENDERECO...: ";E$(Y)
1550 PRINTTAB(10):INPUT"NR.........: ";R$(Y)
1560 PRINTTAB(10):INPUT"TELEFONE...: ";T$(Y)
1570 PRINTTAB(10):INPUT"CIDADE.....: ";C$(Y)
1580 PRINTTAB(10):INPUT"ESTADO.....: ";S$(Y)
1585 GOTO 60
1590 CLS
1600 PRINT:PRINTTAB(10)"NOME.......: ";N$(Y):PRINTTAB(10)"ENDERE
CO...: ";E$(Y);", ";R$(Y):PRINTTAB(10)"FONE.......: ";T$(Y):PRIN
TTAB(10)"CIDADE.....: ";C$(Y);" - ";S$(Y)
1610 PRINT:PRINTTAB(10)"E' ESTE O NOME A ALTERAR ( S/N ) ?"
1620 R$=INKEY$
1630 IF R$=" " GOTO 1620
1640 IF R$="S" GOTO 1520
1650 IF R$="N" GOTO 1490
1660 GOTO 1620
1670 K$=STRING$(17,58)
1680 PRINT:PRINT K$;"C A D A S T R A M E N T O";K$
1690 PRINT@ 4+64,"NOME.......: ";
1700 PRINT@ 6+64,"ENDERECO...: ";
1710 PRINT@ 8+64,"NR.........: ";
1720 PRINT@ 10+64,"TELEFONE...: ";
1730 PRINT@ 12+64,"CIDADE.....: ";
1740 PRINT@ 14+64,"ESTADO.....: ";
1750 RETURN
1760 CLS
1770 Y=0
1780 Y=Y+1
1790 IF N$(Y)="FIM" GOTO 110
1800 GOTO 1780
1810 PRINTTAB(10)"NAO CONSTA DA LISTA":FOR K=1 TO 600:NEXT K:GOT
O 60
```

Lista Telefônica

6 – ***Consulta por telefone:*** para fazer esta consulta, basta fornecer ■ número do telefone: o programa pesquisará e exibirá na tela todos os dados relativos a este número. Esta opção funciona da mesma forma que a opção *Consulta por nome*, inclusive com as mesmas mensagens.

7 – ***Alteração de dados:*** este item permite a alteração de dados durante a digitação, ou mesmo após a gravação dos dados em fita. Na primeira hipótese, será necessário digitar todos os nomes a serem cadastrados, anotando apenas o nome que se quer modificar para, posteriormente, fazer as correções desejadas.

Para modificar, no entanto, os dados já gravados em fita, deve-se, primeiro, utilizar a opção 3 do menu principal e proceder a leitura de todos os dados arquivados em fita. Após a leitura (sempre lembrando de retornar a fita até o início), o programa mostrará a mensagem: **ALTERAÇÕES e QUAL O NOME A ALTERAR.** Digite então o nome a ser modificado, que o programa, ao encontrar o nome solicitado, mostrará na tela todos os dados referentes ao nome pedido e, em seguida, perguntará: **E ESSE O NOME A ALTERAR S/N?.** É preciso atenção ao verificar (comparando-se os dados) se é realmente este o nome a ser corrigido, ou se é somente um homônimo. Se não for o nome que se quer mudar, basta teclar **N** que o programa passará para outro nome igual existente no arquivo. Mas se for o nome a ser corrigido, responda **S** que o vídeo perguntará **NOME?** e o cursor ficará na posição do nome até que se entre com o nome certo.

Supondo-se, entretanto, que o nome que se quer corrigir seja outro, tecle **ENTER** que o nome não será alterado ■ o programa passará para outro item, ■ assim sucessivamente até aparecer o item a ser corrigido. Entre então com os dados corretos correspondentes àquele item e pressione **ENTER** para o cadastro se atualizar. Para retornar ao menu principal, é só continuar apertando **ENTER.**

Este cadastro, por enquanto, somente está atualizado na memória do micro, falta ainda atualizar a fita cassete. Para alterar os dados da fita, pressione a tecla PLAY/RECORD do gravador e entre com a opção 5.

8 – ***Continuar/Arquivo:*** através desta opção pode-se ampliar o número de dados existentes no arquivo. Para acrescentar, por exemplo, mais 20 nomes a um arquivo com 50 nomes já cadastrados, é necessário colocar a fita no gravador e entrar com a opção *Ler arquivo K-7* (3). Depois que o micro terminar de ler todos os dados cadastrados, surgirá na tela o menu principal. Retorne a fita até o início ■ entre com esta opção (8). O programa exibirá o menu de *Cadastramento* e no canto superior direito do vídeo aparecerá o número do nome que será acrescido ao arquivo (neste exemplo, nome nº 51). Depois, com a fita já no início, e ao fim do último nome a ser acrescido no arquivo (neste caso, o vigésimo, que faz o total de 70 nomes no arquivo), deve-se utilizar a opção *Gravar arquivo K-7* (5) que, como já citamos, faz com que o programa retorne ao menu principal e desliga automaticamente o gravador.

LEMBRETES FINAIS

- As opções deste menu que comentamos minuciosamente vão de 1 a 8. Se for digitada uma opção maior ou menor que estas, o programa acusará erro com a mensagem: **OPÇÃO INVÁLIDA.**
- Se, por algum descuido, for digitado **BREAK** no programa, e este ainda tiver dados do cadastro na memória, não tecle **RUN**, pois desta forma todos os dados serão perdidos. Digite **GOTO 60** que os dados não serão afetados. Verifique esta dica pesquisando um nome ou listando o arquivo.
- Verifique se está tudo Ok com o seu programa, testando o programa antes de retirá-lo da memória ■ gravá-lo definitivamente. Para isso, proceda da seguinte forma: após a digitação do programa, grave-o com **CSAVE "LISTA"**; depois da gravação, certifique-se que está tudo certo com **CLOAD? "LISTA"**. Isso é feito retornando a fita com o programa gravado, e o micro, então, compara o programa da memória com o programa da fita. Se tudo estiver correto, após a comparação (LEITURA) surgirá no vídeo a palavra **READY** e logo abaixo o cursor em sua posição normal, mas se tiver ocorrido algum erro, aparecerá na tela a palavra **MAU.**

Divirta-se e teste sua inteligência, neste jogo para a linha Sinclair, mesmo que você esteja...

Solitário

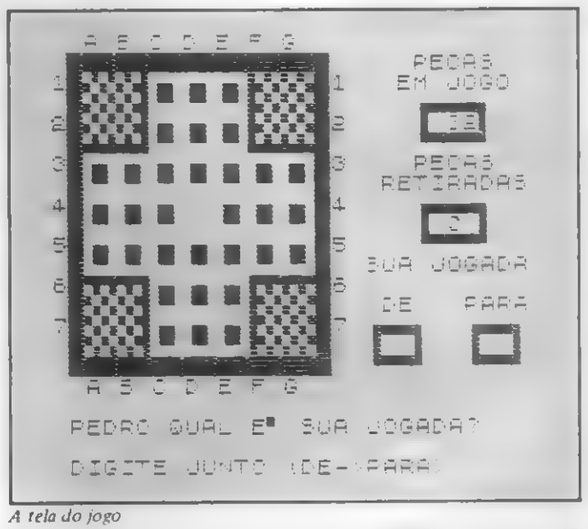
Roberto Ribeiro Peixinho

Como o próprio nome já diz, este jogo é para uma só pessoa. *Solitário* não depende de sorte, sendo baseado em puro raciocínio. Ele tem suas origens na civilização romana, mas só se tornou conhecido durante a Idade Média. A partir daí, espalhou-se por toda a Europa, sob duas versões: uma inglesa e outra francesa. A versão conhecida por nós é a inglesa, com 33 casas e 32 peças.

O programa, para os micros da linha Sinclair, é auto-explicativo, só lembrando aqui que a jogada deve ser feita com as coordenadas juntas (B4D4), seguida de NEWLINE. As jogadas iniciais viáveis são: B4D4; D2D4; F4D4 ou D6D4. Também não se esqueça que quando não existir mais jogadas a fazer, deve-se digitar "00" para saber a classificação.

Agora, paciência e bons lances!

Roberto Ribeiro Peixinho é médico, tem como hobby a computação e, há três meses, trabalha com um TK-85.



A tela do jogo

Solitário

```
  2 REM **SOLITARIO**
  4 REM
  6 REM *ROBERTO R. PEIXINHO*
 10 GOSUB 1050
 20 PRINT
 30 FOR N=1 TO 15
 40 PRINT AT 1 N,"■";AT 16,N;"■"

 50 NEXT N
 60 FOR M=2 TO 15
 70 PRINT AT M,1;"■";AT M,16;"■"

 80 NEXT M
 90 FOR N=2 TO 4
100 FOR M=2 TO 4
110 PRINT AT N,M;"■";AT N,M+11
;"■";AT N+11,M;"■";AT N+11,M+11
;"■"
120 NEXT M
130 NEXT N
140 FOR N=2 TO 5
150 PRINT AT N,5;"■";AT N,12;"■
";AT N+10,5;"■";AT N+10,12;"■"
160 PRINT AT 5,N;"■";AT 5,N+10
;"■";AT 12,N;"■";AT 12,N+10;"■"
170 NEXT N
180 FOR M=2 TO 14 STEP 2
190 PRINT AT 0,M;CHR$ (38+(M-2)
/2);AT 17,M;CHR$ (38+(M-2)/2)
200 PRINT AT M,0;CHR$ (29+(M-2)
/2);AT M,17;CHR$ (29+(M-2)/2)
210 NEXT M
220 DIM B(2)
230 DIM P$(3,2)
240 DIM T$(7)
250 LET P$(1)="■"
260 LET P$(2)="■"
270 LET P$(3)="■"
280 FOR N=6 TO 10 STEP 2
290 FOR M=2 TO 14 STEP 2
300 PRINT AT N,M;P$(1);AT N+1,M
;P$(2)
310 NEXT M
320 NEXT N
330 FOR N=2 TO 4 STEP 2
340 FOR M=6 TO 10 STEP 2
350 PRINT AT N,M;P$(1);AT N+1,M
;P$(2);AT N+10,M;P$(1);AT N+11,M
;P$(2)
360 NEXT M
370 NEXT N
380 PRINT AT 8,8;P$(3);AT 9,8;P
$(3)
390 PRINT AT 1,22;"PEÇAS";TAB 2
1;"EM JOGO"
400 LET P=32
410 PRINT AT 3,22;"■";AT 4,22;"
■";AT 4,24;P;AT 5,22;"■"
420 PRINT AT 8,22;"PEÇAS";TAB 2
```



```
 0 RETIRADAS
 30 PRINT AT 9,22;"■";AT 9,
 24;"■";AT 10,22;"■"
 40 PRINT AT 11,19;"SUA JOGADA
 ";13,20;"DE";AT 13,25;"PARA"
 50 PRINT TAB 19;"■";TAB 19;"■"
 60 LET S(1)=0
 70 LET S(2)=1
 80 FOR I=1 TO 7
 90 LET A(I)=S(2)
100 NEXT I
110 FOR I=1 TO 4: LET A(I)=S(1)
120 NEXT I
130 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
140 INPUT A$
150 FOR U=1 TO 31
160 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
170 INPUT A$
180 IF LEN A$<>4 THEN GOTO 1240
190 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
200 LET L=CODE A$(1)-37
210 LET C=CODE A$(2)-28
220 LET I=CODE A$(3)-37
230 LET D=CODE A$(4)-28
240 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
250 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
260 IF L=I AND C=D THEN GOTO
1000
270 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
280 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
290 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
300 IF C=D+2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
310 IF L=I-2 AND C=D+2 OR L=I+2
AND C=D+2 THEN GOTO 1000
320 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
330 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
340 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
350 IF L=I AND C=D+2 OR L=I AND
C=D-2 THEN GOTO 1000
360 IF L=I+2 AND C=D+2 OR L=I+2
AND C=D-2 THEN GOTO 1000
370 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D-2 THEN GOTO 1000
380 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
390 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
400 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
410 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
420 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
430 LET A=INT ((L+I)/2)
440 LET B=INT ((C+D)/2)
450 IF A(A)=0 THEN GOTO 1000
460 IF B(B)=0 THEN GOTO 1000
470 LET A(A)=0
480 LET B(B)=0
490 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
500 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
510 PRINT TAB 19;"■";TAB 19;"■"
520 LET S(1)=0
530 LET S(2)=1
540 FOR I=1 TO 7
550 LET A(I)=S(2)
560 NEXT I
570 FOR I=1 TO 4: LET A(I)=S(1)
580 NEXT I
590 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
600 INPUT A$
610 FOR U=1 TO 31
620 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
630 INPUT A$
640 IF LEN A$<>4 THEN GOTO 1240
650 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
660 LET L=CODE A$(1)-37
670 LET C=CODE A$(2)-28
680 LET I=CODE A$(3)-37
690 LET D=CODE A$(4)-28
700 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
710 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
720 IF L=I AND C=D THEN GOTO
1000
730 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
740 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
750 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
760 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
770 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
780 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
790 IF L=I AND C=D+2 OR L=I AND
C=D-2 THEN GOTO 1000
800 IF L=I+2 AND C=D+2 OR L=I+2
AND C=D-2 THEN GOTO 1000
810 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D-2 THEN GOTO 1000
820 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
830 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
840 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
850 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
860 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
870 LET A=INT ((L+I)/2)
880 LET B=INT ((C+D)/2)
890 IF A(A)=0 THEN GOTO 1000
900 IF B(B)=0 THEN GOTO 1000
910 LET A(A)=0
920 LET B(B)=0
930 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
940 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
950 PRINT TAB 19;"■";TAB 19;"■"
960 LET S(1)=0
970 LET S(2)=1
980 FOR I=1 TO 7
990 LET A(I)=S(2)
1000 NEXT I
1010 FOR I=1 TO 4: LET A(I)=S(1)
1020 NEXT I
1030 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
1040 INPUT A$
1050 FOR U=1 TO 31
1060 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
1070 INPUT A$
1080 IF LEN A$<>4 THEN GOTO 1240
1090 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
1100 LET L=CODE A$(1)-37
1110 LET C=CODE A$(2)-28
1120 LET I=CODE A$(3)-37
1130 LET D=CODE A$(4)-28
1140 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
1150 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
1160 IF L=I AND C=D THEN GOTO
1000
1170 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
1180 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
1190 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
1200 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
1210 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
1220 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
1230 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
1240 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
1250 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
1260 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
1270 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
1280 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
1290 LET A=INT ((L+I)/2)
1300 LET B=INT ((C+D)/2)
1310 IF A(A)=0 THEN GOTO 1000
1320 IF B(B)=0 THEN GOTO 1000
1330 LET A(A)=0
1340 LET B(B)=0
1350 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
1360 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
1370 PRINT TAB 19;"■";TAB 19;"■"
1380 LET S(1)=0
1390 LET S(2)=1
1400 FOR I=1 TO 7
1410 LET A(I)=S(2)
1420 NEXT I
1430 FOR I=1 TO 4: LET A(I)=S(1)
1440 NEXT I
1450 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
1460 INPUT A$
1470 FOR U=1 TO 31
1480 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
1490 INPUT A$
1500 IF LEN A$<>4 THEN GOTO 1240
1510 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
1520 LET L=CODE A$(1)-37
1530 LET C=CODE A$(2)-28
1540 LET I=CODE A$(3)-37
1550 LET D=CODE A$(4)-28
1560 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
1570 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
1580 IF L=I AND C=D THEN GOTO
1000
1590 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
1600 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
1610 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
1620 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
1630 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
1640 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
1650 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
1660 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
1670 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
1680 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
1690 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
1700 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
1710 LET A=INT ((L+I)/2)
1720 LET B=INT ((C+D)/2)
1730 IF A(A)=0 THEN GOTO 1000
1740 IF B(B)=0 THEN GOTO 1000
1750 LET A(A)=0
1760 LET B(B)=0
1770 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
1780 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
1790 PRINT TAB 19;"■";TAB 19;"■"
1800 LET S(1)=0
1810 LET S(2)=1
1820 FOR I=1 TO 7
1830 LET A(I)=S(2)
1840 NEXT I
1850 FOR I=1 TO 4: LET A(I)=S(1)
1860 NEXT I
1870 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
1880 INPUT A$
1890 FOR U=1 TO 31
1900 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
1910 INPUT A$
1920 IF LEN A$<>4 THEN GOTO 1240
1930 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
1940 LET L=CODE A$(1)-37
1950 LET C=CODE A$(2)-28
1960 LET I=CODE A$(3)-37
1970 LET D=CODE A$(4)-28
1980 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
1990 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
2000 IF L=I AND C=D THEN GOTO
1000
2010 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
2020 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
2030 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
2040 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
2050 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
2060 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
2070 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
2080 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
2090 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
2100 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
2110 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
2120 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
2130 LET A=INT ((L+I)/2)
2140 LET B=INT ((C+D)/2)
2150 IF A(A)=0 THEN GOTO 1000
2160 IF B(B)=0 THEN GOTO 1000
2170 LET A(A)=0
2180 LET B(B)=0
2190 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
2200 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
2210 PRINT TAB 19;"■";TAB 19;"■"
2220 LET S(1)=0
2230 LET S(2)=1
2240 FOR I=1 TO 7
2250 LET A(I)=S(2)
2260 NEXT I
2270 FOR I=1 TO 4: LET A(I)=S(1)
2280 NEXT I
2290 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
2300 INPUT A$
2310 FOR U=1 TO 31
2320 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
2330 INPUT A$
2340 IF LEN A$<>4 THEN GOTO 1240
2350 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
2360 LET L=CODE A$(1)-37
2370 LET C=CODE A$(2)-28
2380 LET I=CODE A$(3)-37
2390 LET D=CODE A$(4)-28
2400 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
2410 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
2420 IF L=I AND C=D THEN GOTO
1000
2430 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
2440 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
2450 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
2460 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
2470 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
2480 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
2490 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
2500 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
2510 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
2520 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
2530 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
2540 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
2550 LET A=INT ((L+I)/2)
2560 LET B=INT ((C+D)/2)
2570 IF A(A)=0 THEN GOTO 1000
2580 IF B(B)=0 THEN GOTO 1000
2590 LET A(A)=0
2600 LET B(B)=0
2610 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
2620 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
2630 PRINT TAB 19;"■";TAB 19;"■"
2640 LET S(1)=0
2650 LET S(2)=1
2660 FOR I=1 TO 7
2670 LET A(I)=S(2)
2680 NEXT I
2690 FOR I=1 TO 4: LET A(I)=S(1)
2700 NEXT I
2710 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
2720 INPUT A$
2730 FOR U=1 TO 31
2740 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
2750 INPUT A$
2760 IF LEN A$<>4 THEN GOTO 1240
2770 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
2780 LET L=CODE A$(1)-37
2790 LET C=CODE A$(2)-28
2800 LET I=CODE A$(3)-37
2810 LET D=CODE A$(4)-28
2820 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
2830 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
2840 IF L=I AND C=D THEN GOTO
1000
2850 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
2860 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
2870 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
2880 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
2890 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
2900 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
2910 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
2920 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
2930 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
2940 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
2950 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
2960 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
2970 LET A=INT ((L+I)/2)
2980 LET B=INT ((C+D)/2)
2990 IF A(A)=0 THEN GOTO 1000
3000 IF B(B)=0 THEN GOTO 1000
3010 LET A(A)=0
3020 LET B(B)=0
3030 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
3040 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
3050 PRINT TAB 19;"■";TAB 19;"■"
3060 LET S(1)=0
3070 LET S(2)=1
3080 FOR I=1 TO 7
3090 LET A(I)=S(2)
3100 NEXT I
3110 FOR I=1 TO 4: LET A(I)=S(1)
3120 NEXT I
3130 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
3140 INPUT A$
3150 FOR U=1 TO 31
3160 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
3170 INPUT A$
3180 IF LEN A$<>4 THEN GOTO 1240
3190 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
3200 LET L=CODE A$(1)-37
3210 LET C=CODE A$(2)-28
3220 LET I=CODE A$(3)-37
3230 LET D=CODE A$(4)-28
3240 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
3250 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
3260 IF L=I AND C=D THEN GOTO
1000
3270 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
3280 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
3290 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
3300 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
3310 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
3320 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
3330 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
3340 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
3350 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
3360 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
3370 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
3380 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
3390 LET A=INT ((L+I)/2)
3400 LET B=INT ((C+D)/2)
3410 IF A(A)=0 THEN GOTO 1000
3420 IF B(B)=0 THEN GOTO 1000
3430 LET A(A)=0
3440 LET B(B)=0
3450 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
3460 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
3470 PRINT TAB 19;"■";TAB 19;"■"
3480 LET S(1)=0
3490 LET S(2)=1
3500 FOR I=1 TO 7
3510 LET A(I)=S(2)
3520 NEXT I
3530 FOR I=1 TO 4: LET A(I)=S(1)
3540 NEXT I
3550 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
3560 INPUT A$
3570 FOR U=1 TO 31
3580 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
3590 INPUT A$
3600 IF LEN A$<>4 THEN GOTO 1240
3610 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
3620 LET L=CODE A$(1)-37
3630 LET C=CODE A$(2)-28
3640 LET I=CODE A$(3)-37
3650 LET D=CODE A$(4)-28
3660 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
3670 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
3680 IF L=I AND C=D THEN GOTO
1000
3690 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
3700 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
3710 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
3720 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
3730 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
3740 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
3750 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
3760 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
3770 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
3780 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
3790 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
3800 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
3810 LET A=INT ((L+I)/2)
3820 LET B=INT ((C+D)/2)
3830 IF A(A)=0 THEN GOTO 1000
3840 IF B(B)=0 THEN GOTO 1000
3850 LET A(A)=0
3860 LET B(B)=0
3870 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
3880 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
3890 PRINT TAB 19;"■";TAB 19;"■"
3900 LET S(1)=0
3910 LET S(2)=1
3920 FOR I=1 TO 7
3930 LET A(I)=S(2)
3940 NEXT I
3950 FOR I=1 TO 4: LET A(I)=S(1)
3960 NEXT I
3970 PRINT AT 10,1;"SEU NOME";A
$;" 10 LETRAS NO MAXIMO"
3980 INPUT A$
3990 FOR U=1 TO 31
4000 PRINT AT 12,1;A$;" QUAL E
A JOGADA"
4010 INPUT A$
4020 IF LEN A$<>4 THEN GOTO 1240
4030 IF A$(1)="0" AND A$(2)="0"
AND A$(3)="0" AND A$(4)="0" THEN
 GOTO 800
4040 LET L=CODE A$(1)-37
4050 LET C=CODE A$(2)-28
4060 LET I=CODE A$(3)-37
4070 LET D=CODE A$(4)-28
4080 PRINT AT 15,20;A$;" ";A$;"
";AT 16,20;A$;" ";A$;"
4090 IF L<1 OR L>7 OR C<1 OR C>
7 OR I<1 OR I>7 OR D<1 OR D>7 THE
N GOTO 1000
4100 IF L=I AND C=D THEN GOTO
1000
4110 IF L=I+2 AND C=D OR I=L+2 AN
D C=D THEN GOTO 1000
4120 IF C=D+2 AND L=I OR C=I+2 AN
D L=I THEN GOTO 1000
4130 IF L=I+2 AND C=D+2 OR I=L+2
AND C=D+2 THEN GOTO 1000
4140 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
4150 IF L=I-2 AND C=D+2 OR L=I-2
AND C=D+2 THEN GOTO 1000
4160 IF C=D-2 AND L=(L+I)/2 INT
(L+I)/2=(C+D)/2 THEN GOTO 1000
4170 IF L=I-2 AND C=D-2 OR L=I+2
AND C=D-2 THEN GOTO 1000
4180 IF (L=I AND C=D-2)+(C=D AND
L=I-2)+(L=I+2 AND C=D-2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
4190 IF (L=I AND C=D+2)+(C=D AND
L=I+2)+(L=I+2 AND C=D+2)+(C=D+2
AND L=I+2)=2 THEN GOTO 1000
4200 IF (L=I AND C=D-2)+(C=D AND
L=I+2)+(L=I+2 AND C=D-2)+(C=D-2
AND L=I+2)=2 THEN GOTO 1000
4210 IF (L=I AND C=D+2)+(C=D AND
L=I-2)+(L=I-2 AND C=D+2)+(C=D+2
AND L=I-2)=2 THEN GOTO 1000
4220 IF L=I AND C=D-2 OR L=I AND
C=D+2 THEN GOTO 1000
4230 LET A=INT ((L+I)/2)
4240 LET B=INT ((C+D)/2)
4250 IF A(A)=0 THEN GOTO 1000
4260 IF B(B)=0 THEN GOTO 1000
4270 LET A(A)=0
4280 LET B(B)=0
4290 PRINT AT 9,22;"■";AT 9,
24;"■";AT 10,22;"■"
4300 PRINT AT 11,19;"SUA JOGADA
";13,20;"DE";AT 13,25;"PARA"
4310 PRINT TAB 19;"■";TAB 19;"■"
4320 LET S(1)=0
4330 LET S(2)=1
4340 FOR I=1 TO 7
4350 LET A(I)=S(2)
4360 NEXT I
4370 FOR I=1 TO 4: LET A(I)=S(1)
4380 NEXT I
4390 PRINT AT 10,1;"SEU NOME";
```

Linha SINCLAIR

Quanto sobra de memória?

Quando você estiver digitando um programa longo e quiser saber exatamente quanto ainda tem de memória disponível, é só entrar com:

```
PRINT(PEEK 16386+256*PEEK 16387)-(
PEEK 16412+256*PEEK 16413)+87
```

Marcio Yamawaki-SP

Linha APPLE

Desafie a velocidade

Aceite este desafio! Veja se consegue ser tão rápido quanto esta dica:

```
10 REM RAPIDO...RAPIDISSIMO
15 REM ARMANDO OSCAR CAVANHA FILHO
20 DIM X(100),Y(100)
30 TEXT:HGR2
35 X(0)=140:Y(0)=80
37 FOR T=1 TO 6:HCOLOR=T
40 FOR I=1 TO 20
50 X(I)=270*RND(1):Y(I)=180*RND(1)
60 HPLOT X(I-1),Y(I-1) TO X(I),Y(I
)
70 NEXT:NEXT:GOTO 30
```

A linha 40 pode ser modificada (sendo que K pode ter qualquer valor até 100) para:

```
40 FOR I=1 TO K
```

E uma boa surpresa para o pessoal que tem TK-2000: para rodar esta dica é só acrescentar esta linha:

```
65 SOUND I*T,3
```

Armando Oscar Cavanha Fº -RJ

Se você tem pequenas rotinas e programas utilitários realmente úteis tomando poeira em seus disquetes ou fitas cassetes, antecipe-se aos piratas e trate de divulgá-los. Envie-os para a REDAÇÃO DE MICRO SISTEMAS – SEÇÃO DICAS: Av. Presidente Wilson, 165/grupo 1210, Centro, Rio de Janeiro, RJ, CEP 20030. Não se esqueça de dizer para qual equipamento foram desenvolvidos. Desta forma, sua descoberta poderá ser útil para muitos e muitos, em vez de desmagnetizar-se com o tempo em suas fitas e disquetes...

Linha TRS-80 COLOR

Efeitos especiais com PCLS

Se você está cansado das cores monótonas do fundo da tela quando usa o PCLS na tela gráfica, tente agora este programinha para mudar o seu panorama visual:

```
10 PMODE 3,1:SCREEN 1,1
20 FOR A=0 TO 255
30 POKE 179,A
40 PCLS
50 NEXT A
```

Observação: o valor A do **POKE** determina o *padrão* colorido do fundo da tela.

Marcos K. Watanabe-SP

Linha TRS-80

Organize seus programas

Aí vai um programinha simples para fornecer uma listagem impressa e devidamente organizada do conteúdo de todos os seus disquetes:

```
10 CLS:PRINT@18,"***IMPRESSAO DE D
IRETORIOS***"
20 PRINT@82,STRING$(29,"-")
30 PRINT@320,"ENTRE O NOME DO DISC
O :":PRINT:LINE INPUT X$
40 CLS:LPRINT"DIRETORIO DO DISCO==
=>   "X$:LPRINT
50 CMD"Z","ON":CMD"D:0":CMD"Z","OF
F":LPRINT:LPRINT:CLS
60 PRINT@655,"TROQUE O DISQUETE NO
 DRIVE 0 E TECLE"
70 PRINT@715,"<ENTER> PARA CONTINU
AR OU <BREAK> PARA PARAR"
80 LINE INPUT X$:GOTO 10
```

E se você possui dois drives e deseja trabalhar no drive 1, basta modificar a linha 50 (**CMD "D:1"**) e a mensagem da linha 60.

Roberto Quito de Sant'Anna-RJ

Linha TRS-80 III

Contagem regressiva

Eis uma boa dica para ser implementada em seus jogos (ou mesmo em programas sérios): uma rotina que faz a contagem regressiva de 9... até 0.

```
6 CLEAR500
7 CLS
8 INPUT"CONTAGEM A PARTIR DE";F:F=
9-F
10 CLS
11 G=9-F
12 FORI=0TOG
15 U$=CHR$(128)
20 A$=CHR$(168):B$=STRING$(4,131):
C$=CHR$(148)
22 X$=CHR$(168):Y$=CHR$(148):Z$=ST
RING$(4,131):Z2$=STRING$(4,131)
23 H$=CHR$(170)
40 CLS
42 F=F+1:IFF=1THENX$=U$:Z2$=U$:GOT
052
43 IFF=2THENGOT052
44 IFF=3THENB$=U$:X$=U$:Z2$=U$:GOT
052
45 IFF=4THENC$=U$:GOTO52
46 IFF=5THENX$=U$:C$=U$:GOTO52
47 IFF=6THENZ2$=U$:X$=U$:Z$=U$:GOT
052
48 IFF=7THENA$=U$:X$=U$:GOTO52
49 IFF=8THENA$=U$:Y$=U$:GOTO52
50 IFF=9THENB$=U$:A$=U$:X$=U$:Z$=U
$:C$=U$:Y$=U$:Z2$=U$:PRINT@412,H$:
PRINT@476,H$;:GOTO57
51 IFF=10THENB$=U$
52 CLS:PRINT@410,A$;:PRINT@415,C$
53 PRINT@474,X$;:PRINT@479,Y$
55 PRINT@475,B$;:PRINT@411,Z$;:PRI
NT@539,Z2$;
57 FORJ=1TO25:OUT255,120+N:OUT255,
121+N:NEXTJ
60 NEXTI
65 CLS:F=0:GOTO8
```

Raimundo Antonio Monteiro-GO

Linha SINCLAIR

Centralizando strings

Crie uma moldura na tela com esta rotina simples que centraliza, rapidamente, strings:

```
1 LET H$="************************
********"
2 INPUT M$
3 LET T=LEN M$
4 LET N=(28-T)/2
5 CLS
6 PRINT AT 9,N;H$( TO T+4)
7 PRINT AT 10,N;"*";TAB(34+T)/2;"*
"
8 PRINT AT 11,N;"*";TAB(32-T)/2;M$
;" *"
9 PRINT AT 13,N;H$;( TO T+4)
```

Marcel Gameleira-AL

Linha TRS-80 COLOR

Aumente a velocidade

Caso você ache que o seu micro compatível com a linha TRS-80 Color não está trabalhando suficientemente rápido, digite então:

```
POKE 65495,0   e  <ENTER>
```

Observe agora que o cursor está piscando com mais velocidade. Para desativar este *high-speed*, basta dar um **RESET** e a velocidade voltará ao normal. Experimente, para testar, rodar um programa – de preferência com muitos cálculos – e cronometrar o tempo gasto para executar as contas. Coloque novamente o programa e digite esta dica. Viu a diferença? Um lembrete importante: nunca tente salvar em fita um programa se o computador estiver em *high-speed*, pois a gravação e o programa na fita irão para o espaço...

Marcos K. Watanabe-SP

Linha SINCLAIR

Arquivando a tela

Coloque em seu micro esta rotina em Assembler, que é dividida em duas partes: a primeira, que vai do endereço 16514 até 16526, executa o armazenamento de uma tela inteira a partir do endereço 30000; e a segunda, que começa no endereço 16527 indo até 16540, que coloca imediatamente no vídeo a tela que foi armazenada.

```
16514   2A 0C 40 11 30 75 01 D6
16522   02 23 ED B0 C9 21 30 75
16530   ED 5B 0C 40 13 01 D6 02
16538   ED B0 C9
```

Adálbero Fernandes Guimarães-MG

TROCO financio ofereco classificados VENDO alugo compro

SOFTWARE

• Soft CP500 (disco), todo tipo troco — Paulo — Cx. P. 6125 — CEP: 13100 — Campinas-SP. Tel.: (0192) 41-8860.

• Vendo ou troco programas para computadores CP 500, CP 300, DGT 100 e similares. Tenho jogos como: Assault, Acrobatas, Star Blazer e outros. 5 mil cada. Faço adaptação de Joystick no CP 300. Paulo Roberto, Rua Sargento João Lopes, 804, Guarabu — Ilha do Governador — RJ. CEP: 21931 — Tel.: 393-7903.

• TK85 e compatíveis. Programas inéditos. Peça relação pelo correio. Bonisoft. Av. Paula e Souza, 422, Maracanã, Rio, RJ. CEP: 20271.

• CPM/Basic ou Cobol. Linha Apple ou outras. Aceito programas objetos para revenda. Tel.: (021) 263-7267, Sérgio/Paulo. Hor. comercial.

• Soft p/Aplle — vendo aplicativos. Tel.: (011) 548-8842.

• Programas p/Apple — os melhores do mercado internacional — 1000 títulos, Cr$ 25.000 disco cheio — Alfamicro — Cx. P. 21193 — SP.

• Programas p/Apple: aplicativos, utilitários, compiladores, linguagens e jogos. Tel.: (021) 239-0449, Stela.

• Programas para Sinclair. Dez por apenas 1 ORTN. Peça catálogo para Softbyte — R. Silvestre Ferraz, 1121 — 37500 Itajubá-MG. Tel.: (035) 622-1602.

• Linha TRS80 Color, 300 programas a sua escolha, peça catálogo, José Luiz Pereira, Cx.P. 1536 — Foz do Iguaçu — CEP: 85890 — PR.

• Commodore-64, assessoria, software, manutenção e acessórios. Av. Brig. Faria Lima, 1644, s/l 26 — São Paulo — SP. Fone: (011) 843-1065.

• Petroclub — Escreva enviando anexo 2000 mil e receba imediatamente jogos e programas para a linha Sinclair ou TK2000, e envie também detalhes do seu micro. Rua Sold. Hercilio Tardeli, 152 — Petrópolis — RJ — CEP: 25600.

• Micro é movido a programa da Microlove. Reabasteça o seu Sinclair e TRS-80 com nossos programas. Peça lista tel.: (011) 448-4372.

• Apple & Compatíveis programas e manuais — solicite listas — Domínio Público Soft & Man — Cx. Postal 201 — S. Bernardo do Campo — CEP: 09700 — São Paulo.

• Vendo software p/todos os micros. Fitas e disco. Tel.: (011) 241-9064, SP.

• Vendo compilador Basic e Forth, editor Assembler, ZX-Debug, Micro Bug, MOS I (25 poderosos comandos) para TK 85/CP 200 Speed, gravados em Eprom Cr$ 60.000 cada. Vendo fita com 10 programas para TK 2000/85 Cr$ 40.000. Gravo Eprom sob encomenda. S. C. Sampaio, Rua Pe. Leopoldo Fernandes, 360 — 60.000, Fortaleza-CE.

• Vendo, troco programas Sinclair TRS80 peça catálogo. Oswaldo Alencar — Av. Gentil Bittencourt, 124/1301 — Belém — Pa. CEP: 66000.

• Compro interpretador Logo, em fita cassete, para TK2000. Preço a combinar. Tel.: 286-1411, tratar com Eduardo.

• Folha de Pgto., Contabilidade, Contr. Estoq., Contas Pag/Rec., Contas Correntes, Faturamento, Banco de Dados, Edit. Textos, Plan. Eletrônica, Utilitários, Geradores de Prog., Copiadores, Manuais, para CP 500 e Apple. Temos também soft para IBM-PC. Despachamos para todo o Brasil, Microservice — O Software Completo. R. Gaspar Fernandes, 16 — São Paulo — Tel.: (011) 215-9283. CEP: 01549.

• Soft p/TK e CP. Lista grátis — FM Software — Cx. P. 85 Tatuí — SP ou R. Proença, 311 — J. Proença — Campinas.

• Vendo programas p/CP500. Tratar c/A. Gaeta Mq. São Vicente, 512/1002 — Rio de Janeiro-RJ.

EQUIPAMENTOS

• Poligames Vídeo & Micro: Toda linha Prológica, Microdigital e Apple. Teclados profissionais p/linha TK e AS-1000. Softhouse programas profissionais p/CP500, jogos e aplicativos p/TK2000 e CP/400. Atendemos reembolso postal. Nossos preços são os melhores, procure-nos para conferir. Rua Cardoso de Moraes, 61, s/loja 311 — Bonsucesso, Rio — Tel.: (021) 270-9197/290-2598.

DIVERSOS

• Manuais em português para micros e periféricos Commodore. Escreva p/W. Belo, R. Itamaracá, 47, D. de Caxias, RJ. ou tel.: (021) 771-6889.

• Vic Commodore manutenção séria, mil soft, manuais em português, cabos, capas, interface K7 e RS 232 para CBBS e projeto Cirandão, manuais técnicos para Epson, TRS 80 e conectores, etc. Bartô Computadores tel.: (021) 262-1213 — Av. Nilo Peçanha, 50/2407 — Rio.

• Traduzo qualquer publicação — Livros, Revistas, Manuais, Artigos em inglês, na área da Microcomputação. Informações com Jorge Eider Silva — Village dos Mares — Quadra 18 — Bloco B — Casa 15 — Capim Macio — 59000 — Natal-RN.

• Compro micros, periféricos, etc. Tel.: (011) 241-9064, SP.

• Apple x Vídeo Texto, Cirandão. Não deixe seu Apple por fora! Temos tudo p/Apple, informações tel.: (011) 241-9064, SP.

• Vendo micros, periféricos, etc. Todas as marcas e modelos. Novos e usados c/garantia. Consulte-nos. Tel.: (011) 241-9064.

• Instalo joystick de videogames compatíveis com Atari p/micros CP200 e CP300. Tratar com Nelson. Tel.: (011) 469-4911 — SP.

• Vendo drives, impressoras, interfaces, expansões, monitores de vídeo, joystick, software, tudo em micros e para micros. Novos e usados c/garantia. Consulte-nos. Tel.: (011) 241-9064, SP.

• Assistência técnica p/micros e periféricos. Todas as marcas. Tel.: (011) 241-9064, SP.

CURSOS

• O NETC — Núcleo de Ensino de Tecnologia e Ciência, estará promovendo a partir do mês de março, cursos inéditos destinados a Técnicos e Engenheiros Eletrônicos, Elétricos e de Telecomunicações. Todos os cursos são ministrados com auxílio de transparências e apostilas, incluindo amplo uso de laboratório de Hardware, contendo entre outros equipamentos, microcomputadores da Suporte Engenharia destinados à treinamento e pesquisa com microprocessadores e periféricos. Alguns exemplos destes cursos são: Eletrônica Digital I, Computadores e Microcomputadores Digitais, Hardware de Sistemas Baseados no Microprocessador Z-80, Hardware de Sistemas Baseados no MP 8080/85, Hardware de Sistemas Baseados no MP 6800, Software do MP Z-80, Software do MP 8080/85, Hardware de Teleprocessamento, Software de Teleprocessamento, Amplificadores Operacionais e Interfaces A/D e D/A. Maiores informações e pedidos de catálogos de cursos pelo telefone: (021) 220-1989, Rua Álvaro Alvim, 37/2º andar — Centro, Rio de Janeiro — RJ. CEP: 20031.

• A Bits e Bytes Computadores ministra regularmente cursos de Basic e DBasic II para computadores da linha CP 300/500 e compatíveis IBM/PC. Aulas em pequenas turmas, também para crianças. Em dias e horário de sua conveniência. Maiores informações: 322-1960 ou 322-2721, Estrada da Gávea, 642, loja B, São Conrado, RJ.

• Mumps — A Matriz promove, mensalmente cursos de Mumps. R. Maestro Elias Lobo, 70 — CEP: 01433, SP. Informações (011) 64-0688.

PARA ANUNCIAR NESTA SEÇÃO ESCREVA PARA:

Av. Presidente Wilson, 165/Grupo 1210
Centro — Rio de Janeiro/RJ — CEP 20030
Tels.: (021) 262-6306
Rua Oliveira Dias, 153 — Jardim Paulista
São Paulo/SP — CEP 01433 — Tels.: (011) 853-3229

M.S. Serviços

ANTMEC

SOFTWARE — CP/M

- Administração Imóveis/Condomínios
- Controle Administrativo/Financeiro p/Clubes, Escolas, Corretoras Seguros
- Controle Operacional Hotéis
- Correção Monetária balanço
- Faturamento Serviços Médicos (Convênios)
- Formulação/Cálculo de Rações
- Gerenciamento Rebanhos Gado Leiteiro e Gado de Corte

Praia de Botafogo nº 210 — C-01
CEP 22250 — Botafogo — RJ
Tel. PBX (021) 551-6699

MACH FORM
MÁQUINAS E FORMULÁRIOS LTDA.

REBOBINAÇÃO DE FITAS DE IMPRESSORAS
ELGIN, DISMAC, ELEBRA e outras
ENCADERNAÇÕES
SERVIÇOS GRÁFICOS
OFF-SET — TIPOGRÁFICO
NOTA FISCAL — FATURA
IMPLANTAÇÃO — RENOVAÇÃO
COMPRA E VENDA DE MÁQUINAS DE ESCRITÓRIO EM GERAL
CONTRATO DE MANUTENÇÃO E CONSERTOS
ARTIGOS DE PAPELARIA

Rua do Propósito, 42 - Sob.
Saúde — R.J.
Tel: (021) 233-1593

ALBAMAR
ELETRÔNICA LTDA.

FITAS CASSETES TAMANHOS
C5 C10 C15 C20
C30 e outros

● FITAS MAGNÉTICAS
1200 e 2400 pés

● DISKETTES
5 1/4 e 8"

Rua Conde de Leopoldina,
270-A São Cristóvão — R.J.
Tels.: (021) 580-6729
580-8276

SBT DADOS

PUBLICAÇÕES TÉCNICAS
Disponíveis em português
1 - Curso de dBase II
2 - Aplicativos dBase II
3 - Relatórios dBase II
4 - Curso de dBase III
5 - Curso de Lotus 1-2-3
6 - Aplicat. Lotus 1-2-3
7 - Curso de Symphony
8 - Curso de Framework
9 - Curso de DOS (PC)
10 - Curso de Unix
11 - Curso de linguagem C
12 - Curso de Wordstar
REG. SOB N.º 0819

VENDAS DISPONÍVEIS PARA TODO BRASIL
Al. Santos, 336 - Cj. 42
CEP 01418 - SP
TEL.: (011) 285-0132

MICROEQUIPO

COMPUTADORES E PERIFÉRICOS
UNITRON
MICROCRAFT
VENDAS
LEASING
PROGRAMAS
CURSOS
ASSISTÊNCIA TÉCNICA

Av. Mal. Câmara, 271 s/loja 101
Tel: (021) 262-3289 — R.J.

PARA PROBLEMAS COM MATERIAL DE
DESENHO - PINTURA - ENGENHARIA
PAPELARIA - ESCRITÓRIO - MÁQUINAS P/ ESCRITÓRIO E SUPRIMENTOS EM GERAL
O BEL-BAZAR ELETRÔNICO
onde você AINDA encontra preço e qualidade de ANTIGAMENTE!
AV. ALMIRANTE BARROSO, 81 - LJ "C"
TEL.: 262-8229 - 262-9068 - 240-8410 - 221-8282
RIO DE JANEIRO - CASTELO

MICROLOGICA
Engenharia de Sistemas Ltda. Consultoria de Hardware

ASSISTÊNCIA TÉCNICA A MICROCOMPUTADORES:

Apple, TRS 80, IBM PC, ZX 81, TK 82, TK 85, CP 200, CP 500, Unitron, Impressoras e demais periféricos
Jogos de xadrez e outros compatíveis.

Compramos seu micro funcionando ou não.

VENDEMOS INTERFACES DIVERSAS P/ LINHA APPLE.

Fazemos transformações e alinhamento de Drivers.

UTILIZE NOSSO CONTRATO PARA ASSISTÊNCIA TÉCNICA

Av. Presidente Vargas, 542/815 — Tel.: 263-9925

MODEMS

ANALÓGICOS - BANDA BASE - SÍNCRONOS - ASSÍNCRONOS

CIRANDÃO EMBRATEL ..... Modelo TS-1275 e TS-300

TROPICAL SISTEMAS

TROPICAL SISTEMAS LTDA.
Av. Antônio Abraão Caran, 430 - 8.º A. - Tel.: (031) 441-1636 - Telex: (031) 1247
Belo Horizonte - Minas Gerais - CEP 30.000

Representantes: Rio - São Paulo - Brasília - Curitiba - Florianópolis - Fortaleza - Maceió - Salvador - Ribeirão Preto - Uberlândia.

Livros

TATCHELL, J. e BENNETT, B., *Introdução ao microcomputador*, **Editora Lutécia.**

Dedicado, segundo pronunciamento da editora Lutécia, a "meninos e meninas de 8 a 16 anos", este livro integra uma série com a qual os editores pretendem ganhar uma fatia do rentável mercado juvenil da Informática.

Todo construído à base de quadrinhos e pequenas legendas, o livro apresenta uma diagramação algo confusa e a ocorrência de definições simplistas e conceitos por vezes imprecisos, o que torna questionável sua eficiência em relação ao leitor-alvo: para o público infantil, ele peca por abordar (geralmente, em duas páginas) temas complexos com o chip, redes locais e controle de processos, e para os adolescentes este tipo de literatura é inconcebível, visto que os jovens na faixa de 15 anos possuem condições de vôos bem mais altos.

Alguns dos temas tratados nos capítulos deste livro são: introdução; o micro; programação; teclado; como executar e guardar programas; gráficos e animações; música e efeitos sonoros; o micro por dentro; uma pastilha por dentro; história do microcomputador; redes de computadores; controle com micros; acessórios para o micro e como escolher seu micro.

Na mesma linha, a Lutécia lançou ainda outros dois títulos da série. *O Guia prático de programação em BASIC*, de Brian Smith, que trata em seus capítulos de assuntos como o funcionamento do computador; dando instruções ao computador; primeiros passos em BASIC; como utilizar o INPUT; o que fazer com o PRINT; desenhos; jogos; loops; subrotinas; gráficos e símbolos e dicas de programação, entre outros temas.

O terceiro livro traz *programas de jogos espaciais*, sendo de autoria de Daniel Isaaman e Jenny Tyler. Neste, encontram-se listagens de programas, já adaptados aos equipamentos nacionais (linhas Sinclair, TRS e Apple), além de sugestões e dicas de programação de jogos. Todos os livros são traduções.

PIAZZI, Pierluigi, *Jogos em Linguagem de Maquina*, **Editora Moderna.**

Jogos em Linguagem de Máquina pode ser considerado uma espécie de antologia de programas para computadores compatíveis com o Sinclair ZX-81. Todos os programas apresentados neste livro são jogos escolhidos segundo vários critérios e não têm o único objetivo de divertir. Embora busque o divertimento, o livro pode ser utilizado como fonte de consulta para atividades técnicas, didáticas e educativas.

Por se tratarem de jogos, os programas contidos no livro exigem uma rapidez de processamento impossível de ser obtida com o BASIC. Assim, a parte essencial de todos os programas é elaborada em ASSEMBLER.

Para tornar a obra acessível a um maior número de pessoas, a parte em linguagem de máquina foi listada de maneira que não se precise conhecer esta linguagem. O primeiro capítulo oferece o programa de um Monitor que permitirá a introdução de códigos no computador de forma bem simples.

Ao final do livro são apresentados dois apêndices: no apêndice A foi listado um programa Monitor para aqueles que possuem um micro com apenas 2Kb de memória RAM; no B foram publicadas algumas cópias da tela para que o leitor possa se organizar melhor, caso queira alterar um display em algum jogo.

MENASCÉ, D.; SCHWABE, D. *Redes de Computadores-Aspectos Técnicos e Operacionais*, **Editora Campus.**

A tecnologia chamada comutação por pacotes tem representado um papel revolucionário na área de comunicação de dados. Isso porque ela permite que o desenvolvimento verificado na área de computação seja diretamente aproveitado na transmissão de dados.

Como resultado desta tecnologia, surgiram as redes de computadores, que formam a base dos modernos sistemas de processamento distribuído.

Assim, este livro trata, inicialmente, dos aspectos de organização de uma rede de computadores, descrevendo, em seguida, protocolos que permitem o seu funcionamento. Além disso, os autores examinam tópicos relacionados às centrais de comutação de pacotes, redes locais de computador e banco de dados distribuídos.

Estes assuntos estão subdivididos do capítulo 2 ao 9 de Redes de Computadores. Nos capítulos 2 e 3 são discutidos os mecanismos básicos usados na organização das redes de computadores e os procedimentos usados para controlar o seu funcionamento. O quarto capítulo apresenta as formas de um computador acessar uma rede, e, no quinto temos as formas de se ligar processos que executam em vários computadores ligados à rede. O capítulo seguinte mostra os protocolos que oferecem serviços do tipo acesso remoto por terminal e transferência de arquivos. No capítulo 7 são discutidas formas de organização dos nós de comutação da rede, enquanto o 8 introduz as redes locais, e o 9 trata de bancos de dados distribuídos. O livro consta, ainda, de um capítulo de conclusões.

MIRSHAWKA, V., *Conhecendo e Utilizando o TK-2000*, **Editora Nobel.**

O objetivo do autor de *Conhecendo e utilizando o TK-2000* é mostrar ao leitor, de maneira didática, as diversas aplicações do TK-2000 Color. Porém, esse objetivo não impediu que a parte prática fosse desenvolvida.

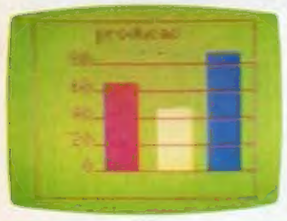
Assim, o autor elaborou e comentou programas para desenvolver problemas de matemática e física; produzir sons que podem se tornar melodias; desenhar figuras reais e abstratas, inclusive gráficos animados; criar desenhos de duas e três dimensões, usando modelos gráficos de baixa e alta resolução, além de programas para o seu divertimento em geral, procurando utilizar quase todas as instruções ou comandos da linguagem BASIC-APPLESOFT. As explicações sobre os comandos utilizados nos programas são dadas à medida que eles aparecem, e a descrição do uso correto dos diversos comandos, embora superficial, é suficiente para que se possa compreendê-los. Ao final de cada capítulo o leitor poderá resolver tarefas que o ajudarão a dominar, ainda mais, o seu TK-2000.

Endereço das Editoras:

- **Editora Campus** — Rua Japeri, 35, Rio Comprido, CEP 20420, Rio de Janeiro, RJ;
- **Editora Lutécia** — Rua Argentina, 171, CEP 20921, Rio de Janeiro, RJ;
- **Editora Moderna** — Rua Afonso Brás, 431, CEP 04511, São Paulo, SP;
- **Editora Nobel** — Rua da Balsa, 559, CEP 02910, São Paulo, SP.

CP400
MICROCOMPUTADOR=COLOR

VOCÊ TEM QUE ESTAR PREPARADO PARA SE DESENVOLVER COM OS NOVOS TEMPOS QUE ESTÃO AÍ. E O CP 400 COLOR É A CHAVE DESSA EVOLUÇÃO PESSOAL E PROFISSIONAL.



POR QUÊ? PORQUE O CP 400 COLOR É UM COMPUTADOR PESSOAL DE TEMPO INTEGRAL: ÚTIL PARA A FAMÍLIA TODA, O DIA INTEIRO.

NA HORA DE SE DIVERTIR, POR EXEMPLO, É MUITO MAIS EMOCIONANTE PORQUE, ALÉM DE OFERECER JOGOS INÉDITOS, É O ÚNICO COM 2 JOYSTICKS ANALÓGICOS DE ALTA SENSIBILIDADE, QUE PERMITEM MOVIMENTAR AS IMAGENS EM TODAS AS DIREÇÕES, MESMO. NA HORA DE TRABALHAR E ESTUDAR, O CP 400 COLOR MOSTRA O SEU LADO SÉRIO: MEMÓRIA EXPANSÍVEL, PORTA PARA COMUNICAÇÃO DE DADOS, SAÍDA PARA IMPRESSORA, E UMA ÓTIMA NITIDEZ COM IMAGENS COLORIDAS.

COMO SE TUDO ISSO NÃO BASTASSE, A PROLÓGICA AINDA OFERECE A GARANTIA DE QUALIDADE DE QUEM É LÍDER NA TECNOLOGIA DE COMPUTADORES, E O PREÇO MAIS ACESSÍVEL NA CATEGORIA.

NUMA FRASE: SE VOCÊ NÃO QUISER CHEGAR ATRASADO AO FUTURO, COMPRE SEU CP 400 COLOR IMEDIATAMENTE.

EMOÇÃO E INTELIGÊNCIA NUM EQUIPAMENTO SÓ.

- MICROPROCESSADOR: 6809E COM ESTRUTURA INTERNA DE 16 BITS E CLOCK DE FREQÜÊNCIA DE ATÉ 1.6 MHZ.
- POSSIBILITA O USO DE ATÉ 9 CORES, E TEM UMA RESOLUÇÃO GRÁFICA SUPERIOR A 49.000 PONTOS.
- MEMÓRIA ROM: 16K BYTES PARA SISTEMA OPERACIONAL E INTERPRETADOR BASIC.
- MEMÓRIA RAM: O CP 400 COLOR ESTÁ DISPONÍVEL EM DOIS MODELOS:

  - MODELO 16K: EXPANSÍVEL A 64K BYTES.
  - MODELO 64K: ATÉ 64K BYTES QUANDO USADO COM O NOVO DISK-SYSTEM, CP 450.
- O CP 400 COLOR DISPÕE DE CARTUCHOS DE PROGRAMAS COM 16K BYTES DE CAPACIDADE, QUE PERMITEM O CARREGAMENTO INSTANTÂNEO DE JOGOS, LINGUAGENS E APLICATIVOS COMO: BANCO DE DADOS, PLANILHAS DE CÁLCULO, EDITORES DE TEXTOS, APLICATIVOS FINANCEIROS, APLICATIVOS GRÁFICOS, ETC.
- SAÍDA SERIAL RS 232 C QUE PERMITE COMUNICAÇÃO DE DADOS. ALÉM DO QUE, ATRAVÉS DESTA PORTA, VOCÊ PODE CONECTAR QUALQUER IMPRESSORA SERIAL OU ATÉ MESMO FORMAR UMA REDE DE TRABALHO COM OUTROS MICROS.

- PORTA PARA GRAVADOR CASSETE COM GRAVAÇÃO E LEITURA DE ALTA VELOCIDADE.
- SAÍDAS PARA TV EM CORES E MONITOR PROFISSIONAL.
- DUAS ENTRADAS PARA JOYSTICKS ANALÓGICOS QUE OFERECEM INFINITAS POSIÇÕES NA TELA, ENQUANTO OUTROS TÊM SOMENTE 8 DIREÇÕES.
- AMPLA BIBLIOTECA DE SOFTWARE JÁ DISPONÍVEL.
- ALIMENTAÇÃO: 110-220 VOLTS.

VEJA, TESTE E COMPRE SEU CP 400 COLOR NOS MAGAZINES E REVENDEDORES PROLÓGICA.

TECNOLOGIA PROLOGICA

CP
COMPUTADORES PESSOAIS

RUA PTOLOMEU, 650 - VILA SOCORRO
SÃO PAULO, S.P. - CEP 04762
FONES: (PBX)523-9609/548-0749/548-4540

QUEM TEM UM, TEM FUTURO.

Apresentamos o TK 2000 II. Ele roda o programa mais famoso do mundo.

De hoje em diante nenhuma empresa, por menor que seja, pode dispensar o TK 2000 II. Por que?

O novo TK 2000 II roda o Multicalc: a versão Microsoft do Visicalc®, o programa mais famoso em todo o mundo.

Isto significa que, com ele, você controla estoques, custos, contas a pagar, faz sua programação financeira, efetua a folha de pagamentos e administra minuto a minuto as suas atividades.

Detalhe importante: o novo TK 2000 II, com Multicalc, pode intercambiar planilhas com computadores da linha Apple®.

E, como todo business computer que se preza, ele tem teclado profissional, aceita monitor, diskette, impressora e já vem com interface.

Além de poder ser ligado ao seu televisor (cores ou P&B), oferecendo som e imagem da melhor qualidade.

Portanto, peça logo uma demonstração do novo TK 2000 II, nas versões 64K ou 128K de memória.

A mais nova estrela do show business só espera por isto para estrear no seu negócio.

Preço de lançamento* (128 K):
Cr$ 2.649.850

MICRODIGITAL
computadores pessoais

Open for Business.

* Sujeito a alteração sem prévio aviso.